



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ**  
**ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

**Ανθρώπινος εγκέφαλος και βιο-εμπνευσμένοι αλγόριθμοι**

**Σοβατζίδα Γεωργία**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**Υπεύθυνος**  
**Δημήτριος Ιακωβίδης**  
**Αναπληρωτής Καθηγητής**

**Λαμία, 2019**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ  
ΒΙΟΙΑΤΡΙΚΗ**

**Ανθρώπινος εγκέφαλος και βιο-εμπνευσμένοι αλγόριθμοι**

**Σοβατζίδα Γεωργία**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων  
Δημήτριος Ιακωβίδης  
Αναπληρωτής Καθηγητής**

**Λαμία, 2019**

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις <sup>(1)</sup>, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ...../...../20.....

Ο – Η Δηλ.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

**Ανθρώπινος εγκέφαλος και βιο-εμπνευσμένοι αλγόριθμοι**

**Σοβατζίδα Γεωργία**

**Τριμελής Επιτροπή:**

Δημήτριος Ιακωβίδης, Αναπληρωτής Καθηγητής (επιβλέπων)

Κωνσταντίνος Δελήμπασης, Αναπληρωτής Καθηγητής

Αθανάσιος Λουκόπουλος, Επίκουρος Καθηγητής



## ΠΕΡΙΛΗΨΗ

Οι αλγόριθμοι κατέχουν κυρίαρχο ρόλο στη σύγχρονη ζωή, γι' αυτό και τα τελευταία χρόνια δημιουργούνται όλο και περισσότεροι. Αρκετοί από αυτούς είναι εμπνευσμένοι από τη φύση και βασίζονται σε επιτυχημένα χαρακτηριστικά του βιολογικού συστήματος. Οι αλγόριθμοι αυτοί ονομάζονται βιολογικά-εμπνευσμένα ή βιο-εμπνευσμένοι. Μία ειδικότερη κατηγορία βιο-εμπνευσμένων αλγορίθμων, απαρτίζουν οι αλγόριθμοι που αντλούν την πηγή έμπνευσής τους από τον εγκέφαλο. Ο εγκέφαλος, όπως είναι γνωστό, είναι ένα από τα σημαντικότερα όργανα του σώματος, καθώς ρυθμίζει βασικές λειτουργίες ολόκληρου του οργανισμού. Μία λοιπόν από τις χρήσιμες αυτές λειτουργίες, είναι η παραγωγή ιδεών, μία διαδικασία που λέγεται ιδεοκαταιγισμός (brainstorming). Στον ιδεοκαταιγισμό συγκεντρώνονται κάποια άτομα, συζητούν, παράγουν και ανταλλάσσουν ιδέες για ένα πρόβλημα, με σκοπό να καταλήξουν στην εύρεση της καλύτερης και πιο αποτελεσματικής λύσης. Η διαδικασία αυτή αποτέλεσε πηγή έμπνευσης για να δημιουργηθεί ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών, που μελετάει η παρούσα πτυχιακή εργασία. Πιο αναλυτικά, αυτή η εργασία έχει ως στόχο τη διερεύνηση αλγορίθμων που είναι εμπνευσμένοι από τη βιολογία και ειδικότερα από τον εγκέφαλο και τις λειτουργίες του, με σκοπό την εύρεση ενός συγκεκριμένου αλγορίθμου, σχετικά πρόσφατου και την πρωτότυπη εφαρμογή του, ώστε να εξεταστεί η χρησιμότητά του. Αφού εντοπίζεται και περιγράφεται αναλυτικά ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών, εξετάζεται πειραματικά σε τρεις διαφορετικές πρωτότυπες εφαρμογές. Έτσι, στην πρώτη εφαρμογή, ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών χρησιμοποιείται για κατηγοριοποίηση δεδομένων στις Μηχανές Διανυσμάτων Στήριξης, με στόχο την εύρεση των βέλτιστων παραμέτρων που θα επιφέρουν το ελάχιστο μέσο τετραγωνικό σφάλμα. Στη συνέχεια, ο αλγόριθμος εφαρμόζεται για αποθορυβοποίηση εικόνων, μέσω ελαχιστοποίησης της τιμής του μέσου τετραγωνικού σφάλματος, με χρήση του φίλτρου Ενδιάμεσης Τιμής, καθώς επίσης και για αποθορυβοποίηση εικόνων με χρήση Διδιάστατου Διακριτού Μετασχηματισμού Κυματιδίου. Η βέλτιστη τιμή PSNR (*Peak to Noise Ratio*) ισούται με 34.5 και επιτυγχάνεται, με χρήση φίλτρου ενδιάμεσης τιμής. Όπως φαίνεται και από τα πειραματικά αποτελέσματα, ο αλγόριθμος δείχνει να έχει πολλές μελλοντικές προοπτικές, καθώς επιφέρει ιδιαίτερα θετικά αποτελέσματα, συγκριτικά με άλλους αλγορίθμους.

# ABSTRACT

Algorithms have a dominant role in modern life, so more and more are of them are being created in recent years. Several of them are inspired by nature and are based on successful characteristics of the biological system. These algorithms are called biologically-inspired or bio-inspired. A more specific category of bio-inspired algorithms, consists of the algorithms that derive the source of their of inspiration from the brain. The brain, as it is known, is one of the most important organs in the body, as it regulates basic functions of the whole organism. So one of these useful functions is the production of ideas, a process called brainstorming. In the brainstorming process, people gather, discuss, produce and exchange ideas for a problem, in order to find the best and most effective solution. This process was a source of inspiration for creating the brainstorming algorithm that studies this thesis. In detail, this work is aimed at investigating algorithms inspired by biology and especially by the brain and its functions, in order to find a specific algorithm, relatively recent and some unusual applications in order to examine its usefulness. Once the brainstorming algorithm is identified and described in detail, it is examined experimentally in three different applications. Thus, in the first application, the brainstorming algorithm is used to categorize data into the support vector machines, in order to find the optimal parameters that will bring the minimum mean square error. The algorithm is then applied for noise reduction, by minimizing the value of the average square error, using the median filter, as well as for noise-proofing images using 2D Discrete Wavelet Transform. The best and optimized PSNR (Peak to Noise Ratio) value is 34.5 and is achieved by applying median filter. As illustrated by the experimental results, the algorithm seems to have many future perspectives, as it results in very positive outcomes, compared to other algorithms.







# ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος εικόνων .....	6
Κατάλογος πινάκων.....	7
1. Εισαγωγή.....	9
1.2 Στόχοι .....	9
1.3 Δομή Εργασίας.....	9
2.Βιολογικά και Τεχνητά Νευρωνικά Δίκτυα .....	11
2.1 Εγκέφαλος και Βιολογικά Νευρωνικά Δίκτυα .....	11
2.1.1Η ανατομία του εγκεφάλου .....	12
2.1.2 Η νευροφυσιολογία του εγκεφάλου .....	13
2.2Τεχνητά Νευρωνικά Δίκτυα .....	14
2.2.1Μοντέλο Τεχνητού Νευρώνα .....	14
2.2.2Πλεονεκτήματα Νευρωνικών Δικτύων .....	18
2.2.3Αλγόριθμος Ανάστροφης Μετάδοσης Σφάλματος.....	18
2.2.4 Σύγχρονες εκδοχές νευρωνικών δικτύων .....	22
2.2.5Αλγόριθμοι εκπαίδευσης ΤΝΔ .....	23
3.Βιο-εμπνευσμένοι αλγόριθμοι.....	26
3.2 Μεθοδολογία Βιβλιογραφικής Αναζήτησης.....	26
3.3 Αποτελέσματα Βιβλιογραφικής Αναζήτησης.....	27
3.4 Καταγραφή και Ομαδοποίηση Αλγορίθμων.....	30
4.Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών .....	43
4.1 Η διαδικασία καταιγισμού ιδεών.....	43
4.2 Κανόνες του Osborn.....	44
4.3 Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών .....	45
4.3.1 Ομαδοποίηση λύσεων .....	45
4.3.2 Παραγωγή νέων ιδεών.....	46
4.3.3 Επιλογή ιδεών.....	46
4.4 Βιβλιογραφική ανασκόπηση & Εφαρμογές του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών.....	50

5.Καταιγισμός Ιδεών για τη Βελτιστοποίηση Παραμέτρων Μηχανών Διανυσμάτων Στήριξης.....	64
5.1Γραμμική Κατηγοριοποίηση .....	65
5.1.1 Γραμμικά Διαχωρίσιμη Κατηγοριοποίηση.....	65
5.1.2 Γραμμικά Μη Διαχωρίσιμη Κατηγοριοποίηση .....	66
5.2 Μη Γραμμική Κατηγοριοποίηση.....	68
5.3Συνθήκη του Mercer.....	68
5.4Συναρτήσεις πυρήνα.....	68
5.5Αλγόριθμος Καταιγισμού Ιδεών και κατηγοριοποίηση δεδομένων .....	69
6.Βελτιστοποίηση Καταιγισμού Ιδεών για τη Μείωση του Θορύβου σε Εικόνες .....	75
6.2 Θόρυβος στις εικόνες .....	75
6.3Αλγόριθμος καταιγισμού ιδεών και αποθορυβοποίηση εικόνων .....	76
6.3.1 Αποθορυβοποίηση εικόνων με φίλτρο ενδιάμεσης τιμής.....	76
6.3.2 Αποθορυβοποίηση εικόνων με χρήση μετασχηματισμού κυματιδίου.....	82
7.Συμπεράσματα.....	90
Βιβλιογραφία .....	92

## Κατάλογος εικόνων

Εικόνα 1 Προστατευτικά περιβλήματα εγκεφάλου.....	12
Εικόνα 2 Οι λοβοί του εγκεφάλου (Gray, 1878).....	13
Εικόνα 3 Σχηματική αναπαράσταση μη γραμμικού νευρώνα. Περιλαμβάνει τα συναπτικά βάρη (synaptic weights), έναν αθροιστή (summing junction), τη συνάρτηση ενεργοποίησης (activation function) και το κατώφλι (threshold).....	16
Εικόνα 4Βηματική συνάρτηση για την τιμή κατωφλίου T.....	16
Εικόνα 5 Βιολογικός νευρώνας.....	17
Εικόνα 6 Διαδικασία Εκπαίδευσης Ανάστροφης Μετάδοσης Σφάλματος (Βλαχαβας, 2006).....	22
Εικόνα 7 Πλήθος δημοσιεύσεων ανά κατηγορία βιο-εμπνευσμένου αλγορίθμου στις βάσεις δεδομένων Scopus, Google Scholar .....	30
Εικόνα 8Ομαδοποίηση βιο-εμπνευσμένων αλγορίθμων .....	31
Εικόνα 9 Διάγραμμα Ροής Αλγορίθμου BSO .....	49
Εικόνα 10 Πλήθος δημοσιεύσεων Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών. Πηγή: Scopus, Google Scholar .....	63
Εικόνα 11Γραμμικά Διαχωρίσιμη Κατηγοριοποίηση (Burges, 1998b) .....	65

Εικόνα 12 Γραμμικά Μη Διαχωρίσιμη Κατηγοριοποίηση(Burges, 1998b).....	67
Εικόνα 13 Απεικόνιση της γραμμικά διαχωρίσιμης (αριστερά) και μη διαχωρίσιμης (δεξιά) ταξινόμησης(Burges, 1998b).....	67
Εικόνα 14 Διάγραμμα Ροής Συσχετίσεων για Μηχανές Διανυσμάτων Στήριξης.....	70
Εικόνα 15 Συνοπτικό διάγραμμα εφαρμογής 1 BSO-SVM.....	71
Εικόνα 16 Διάγραμμα αποθορυβοποίησης εικόνας με τονBSO.....	76
Εικόνα 17 Αρχική εικόνα.....	78
Εικόνα 18(α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής $5 \times 3$ και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	78
Εικόνα 19 Αρχική εικόνα.....	79
Εικόνα 20 (α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής $3 \times 3$ και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	79
Εικόνα 21 Αρχική εικόνα.....	81
Εικόνα 22(α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής $5 \times 3$ και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	81
Εικόνα 23 Ο διδιάστατος διακριτός μετασχηματισμός κυματιδίου τριών επιπέδων.....	82
Εικόνα 24 Βασικά βήματα Διδιάστατου Διακριτού Μετασχηματισμού Κυματιδίου αποσύνθεσης εικόνας.....	83
Εικόνα 25 Αρχική εικόνα.....	85
Εικόνα 26 (α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	86
Εικόνα 27 Αρχική εικόνα.....	87
Εικόνα 28 (α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	87
Εικόνα 29 Αρχική εικόνα.....	88
Εικόνα 30 (α) Θορυβοποιημένη εικόνα (αριστερά)   (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά).....	88

## Κατάλογος πινάκων

Πίνακας 1 Κατηγορίες αλγορίθμων.....	31
--------------------------------------	----

Πίνακας 2 Είδη και εφαρμογές Βιο-εμπνευσμένων Αλγορίθμων (2016-2017) .....	33
Πίνακας 3 Τα βήματα διεξαγωγής της διαδικασίας καταγισμού ιδεών .....	44
Πίνακας 4 Αρχικοποίηση μεταβλητών BSO για SVM.....	47
Πίνακας 5 Βήματα υλοποίησης αλγορίθμου BSO .....	48
Πίνακας 6 Εφαρμογές BSO (2011-2017).....	62
Πίνακας 7 Περιγραφή δεδομένων από το UCI.....	72
Πίνακας 8 Αποτελέσματα υπολογισμού μέσης τιμής και τυπικής απόκλισης μέσου τετραγωνικού σφάλματος με τη βοήθεια βιο-εμπνευσμένων αλγορίθμων.....	73
Πίνακας 9 Διάγραμμα μέσης τιμής μέσου τετραγωνικού σφάλματος συσχετισμένο με τη τυπική απόκλιση.....	74
Πίνακας 10 Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής $N \times M$ στην εικόνα της Lena .....	78
Πίνακας 11 Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής $N \times M$ στην εικόνα Cameraman .....	79
Πίνακας 12 Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής $N \times M$ στην εικόνα Peppers.....	81
Πίνακας 13 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με hard threshold και soft threshold και διαφορετικά επίπεδα θορύβου, της εικόνας Lena. ....	86
Πίνακας 14 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Multi-Objective PSO, με hard threshold και soft threshold, της εικόνας Lena. (Sivaranjanietal, 2019)86	
Πίνακας 15 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με hardthreshold και softthreshold και διαφορετικά επίπεδα θορύβου, της εικόνας cameran.....	87
Πίνακας 16 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με hardthreshold και softthreshold και διαφορετικά επίπεδα θορύβου, της εικόνας Cameran.....	88
Πίνακας 17 Βέλτιστες τιμές PSNR, συγκεντρωτικά, για αποθορυβοποίηση εικόνων με εφαρμογή του BSO Αλγορίθμου.....	89

# 1. Εισαγωγή

Ο ανθρώπινος εγκέφαλος είναι σαν το σύμπαν, καθώς απαρτίζεται από τόσα δισεκατομμύρια νευρώνες όσοι και τα αστέρια. Οι διεργασίες που επιτελεί ο εγκέφαλος είναι ιδιαίτερα σημαντικές και ελκύουν το ενδιαφέρον ατόμων από πολλούς κλάδους και επιστήμες όπως γιατρούς, καθηγητές, φιλοσόφους, ψυχολόγους και ερευνητές.

Η έρευνα της παρούσας πτυχιακής εργασίας εστιάζει στη διερεύνηση βιο-εμπνευσμένων αλγορίθμων εστιάζοντας περισσότερο σε αλγορίθμους εμπνευσμένους από τον ανθρώπινο εγκέφαλο και τον τρόπο με τον οποίο λειτουργεί. Μελετώνται επίσης εφαρμογές αναφορικά με την κατηγοριοποίηση δεδομένων και επεξεργασία εικόνων.

## 1.2 Στόχοι

Καθημερινά δημιουργούνται καινούριοι αλγόριθμοι και η πηγή έμπνευσής τους μπορεί να προέρχεται από διάφορους τομείς, όπως η φύση, επιστήμες όπως η φυσική, η χημεία και η βιολογία, ακόμα και διεργασίες του ανθρώπινου σώματος, όπως η λειτουργία του εγκεφάλου. Οι βιο-εμπνευσμένοι αλγόριθμοι, τους οποίους μελετά η συγκεκριμένη εργασία, αντλούν την πηγή έμπνευσής τους από τη βιολογία.

Οι στόχοι της παρούσας πτυχιακής εργασίας μπορούν να συνοψιστούν στα ακόλουθα σημεία:

- Διερεύνηση βιο-εμπνευσμένων αλγορίθμων
- Επιλογή κατάλληλου αλγορίθμου, έπειτα από αναζήτηση, εμπνευσμένο από τον εγκέφαλο
- Πειραματική εφαρμογή αλγορίθμου για κατηγοριοποίηση δεδομένων
- Πειραματική εφαρμογή αλγορίθμου για αποθορυβοποίηση εικόνων
- Αξιολόγηση πειραμάτων και εξαγωγή συμπερασμάτων

## 1.3 Δομή Εργασίας

Η συνέχεια της εργασίας είναι οργανωμένη σε επτά ενότητες:

Στην **Ενότητα 2** παρουσιάζονται από θεωρητική προσέγγιση τα βιολογικά και τεχνητά νευρωνικά δίκτυα. Πιο συγκεκριμένα, η ενότητα αυτή χωρίζεται σε περαιτέρω υποενότητες που αφορούν την ανατομία και τη φυσιολογία του εγκεφάλου, ενώ όσον αφορά τα τεχνητά νευρωνικά δίκτυα περιγράφεται το μοντέλο του τεχνητού νευρώνα, τα πλεονεκτήματα των νευρωνικών δικτύων, ο αλγόριθμος εκπαίδευσης ανάστροφης μετάδοσης λάθους, σύγχρονες εκδοχές νευρωνικών δικτύων και τέλος αλγόριθμοι που χρησιμοποιούνται για την εκπαίδευση των τεχνητών νευρωνικών δικτύων.

Στην **Ενότητα 3** αναλύονται οι βιο-εμπνευσμένοι αλγόριθμοι. Η ενότητα αυτή περιγράφει τη μεθοδολογία αναζήτησης των αλγορίθμων, που ακολουθήθηκε, προκειμένου να επιλεγεί ένας συγκεκριμένος αλγόριθμος για μελέτη και πειραματική εφαρμογή. Παρουσιάζονται, ακόμα, τα αποτελέσματα των αναζητήσεων, καθώς και η βιβλιογραφική ανασκόπηση που πραγματοποιήθηκαν.

Στην **Ενότητα 4** περιγράφεται αναλυτικά ο Αλγόριθμος Καταιγισμού Ιδεών (Brain Storm Optimization Algorithm (BSO)). Η ενότητα αυτή χωρίζεται σε επιμέρους υποενότητες, που αναφέρονται στην περιγραφή της διαδικασίας καταιγισμού ιδεών, τους κανόνες του Osborn, την περιγραφή του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών, καθώς και τη βιβλιογραφική ανασκόπηση που πραγματοποιήθηκε για τις εφαρμογές του συγκεκριμένου αλγορίθμου.

Στην **Ενότητα 5** παρουσιάζονται οι Μηχανές Διανυσμάτων Στήριξης (Support Vector Machines (SVM)). Πιο συγκεκριμένα, οι υποενότητες, που περιέχονται στην ενότητα αυτή, αναφέρονται στη γραμμική κατηγοριοποίηση, τη μη γραμμική κατηγοριοποίηση, τη συνθήκη του Mercer, τις συναρτήσεις πυρήνα και σε εφαρμογές του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών για κατηγοριοποίηση δεδομένων.

Στην **Ενότητα 6** παρουσιάζεται η επεξεργασία εικόνας. Πιο συγκεκριμένα, η ενότητα αυτή χωρίζεται σε υποενότητες που αφορούν το θόρυβο στις εικόνες, την αποθορυβοποίηση εικόνων με φίλτρα, εφαρμογές του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών για αποθορυβοποίηση εικόνων με φίλτρα ενδιάμεσης τιμής και εφαρμογές του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών για αποθορυβοποίηση εικόνων με τη χρήση μετασχηματισμού κυματιδίου.

Τέλος, στην **Ενότητα 7** παρουσιάζονται τα συμπεράσματα που προκύπτουν από την παρούσα πτυχιακή εργασία, καθώς και μελλοντικές ερευνητικές κατευθύνσεις.



## 2.Βιολογικά και Τεχνητά Νευρωνικά Δίκτυα

Το νευρωνικό δίκτυο είναι ένα δίκτυο που αποτελείται από απλούς υπολογιστικούς διασυνδεδεμένους νευρώνες. Δημιουργήθηκε εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο προσπαθεί και να προσομοιώσει. (Haykin, 1999; Διαμανταρας, 2007) Τα εκατομμύρια νευρώνες που αποτελείται ο εγκέφαλος, έχουν ως σκοπό να εκτελούν όλες τις βασικές και μη λειτουργίες του ανθρώπινου οργανισμού όπως είναι η όραση, η κίνηση και η σωστή απόκριση σε ερεθίσματα. Βασική αρχή της επιστήμης αυτής βασίστηκε στο γεγονός ότι ο ανθρώπινος εγκέφαλος εκτελεί τους υπολογισμούς με τελείως διαφορετικό τρόπο από έναν συμβατικό υπολογιστή.

Υπάρχουν οι βιολογικοί νευρώνες, οι οποίοι αναφέρονται σε συγκεκριμένα τμήματα νευρικού ιστού, όμως, υπάρχουν επίσης και οι τεχνητοί νευρώνες οι οποίοι αποτελούν αλγοριθμικό κατασκευάσμα. Στόχος τους είναι η επίλυση κάποιου προβλήματος μαθηματικής- υπολογιστικής φύσεως ή ακόμα και η προσομοίωση λειτουργιών με χρήση συγκεκριμένου και γνωστού μαθηματικού μοντέλου. Παρακάτω περιγράφονται ο εγκέφαλος, η φυσιολογία αυτού καθώς και τα βιολογικά νευρωνικά δίκτυα, οι πηγές των οποίων είναι οι: (Snell, 2010), (Faiz, Blackburn, & Moffat, 2011) (Stufflebeam, 2008)(Ward & Linden, 2017) καθώς και (Βλαχαβας, 2006).

### 2.1 Εγκέφαλος και Βιολογικά Νευρωνικά Δίκτυα

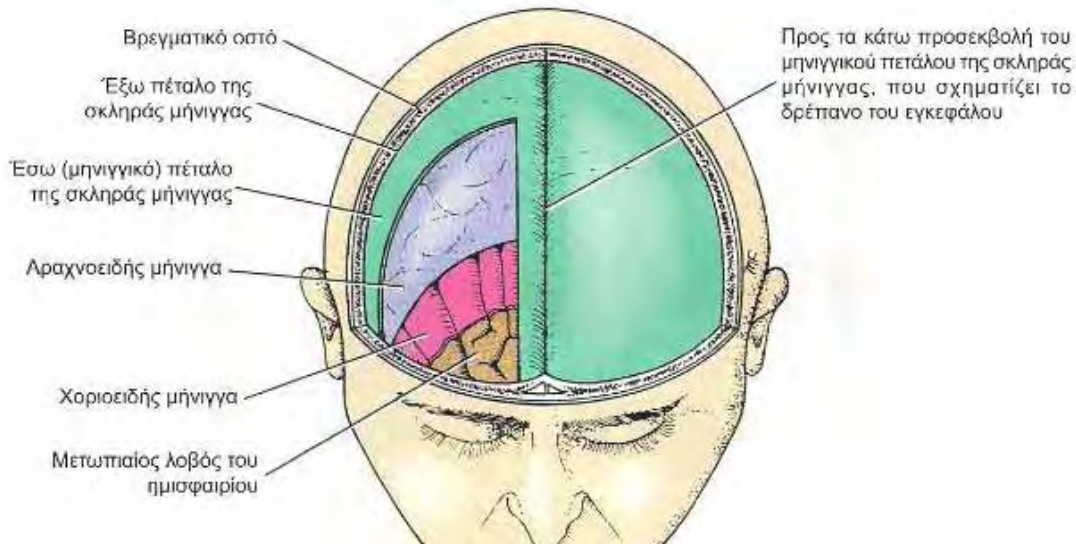
Το νευρικό σύστημα και το σύστημα ενδοκρινών αδένων ελέγχουν τις λειτουργίες του ανθρώπινου σώματος. Το νευρικό σύστημα αποτελείται κυρίως από εξειδικευμένα κύτταρα, των οποίων η λειτουργία είναι να υποδέχονται αισθητικά ερεθίσματα και να τα μεταφέρουν στα εκτελεστικά όργανα, δηλαδή τους μυς και τους αδένες. Το νευρικό σύστημα υποδιαιρείται σε δύο κύρια μέρη: το Κεντρικό Νευρικό Σύστημα, το οποίο αποτελείται από τον εγκέφαλο και το νωτιαίο μυελό (*spinal cord*), και το περιφερικό νευρικό σύστημα, το οποίο αποτελείται από τα *εγκεφαλικά νεύρα (cerebral nerves)* και τα νωτιαία νεύρα (*spinal nerves*) με τα νευρικά γάγγλια (*spinal ganglion*) τους.

Στο κεντρικό νευρικό σύστημα ο εγκέφαλος και ο νωτιαίος μυελός είναι τα κύρια κέντρα όπου επεξεργάζονται οι νευρικές πληροφορίες. Τόσο ο εγκέφαλος όσο και ο νωτιαίος μυελός επικαλύπτονται με μεμβράνες, τις μήνιγγες (*brain mantles/maters*), περιβάλλονται εξωτερικά από το εγκεφαλονωτιαίο υγρό (*cerebrospinal fluid (CSF)/ spinal fluid*) και προστατεύονται επιπρόσθετα από τα οστά του κρανίου και της σπονδυλικής στήλης.

Το κεντρικό νευρικό σύστημα εσωτερικά, είναι οργανωμένο σε φαιά και σε λευκή ουσία. Η φαιά ουσία (*cinerea/grey matter*) αποτελείται από τα νευρικά κύτταρα και τμήματα των αποφυάδων τους, περιβαλλόμενα από νευρογλοία (*neuroglia*). Η λευκή ουσία (*materia alba/matter white*), αποτελείται από νευρικές ίνες περιβαλλόμενες από νευρογλοία. Έχει λευκό χρώμα λόγω μίας λιποειδούς ουσίας που βρίσκεται στο μυελώδες έλυτρο πολλών νευρικών ινών. Σε αντίθεση προς το νωτιαίο μυελό, ο εγκέφαλος αποτελείται από μία εσωτερική μάζα λευκής ουσίας η οποία περιβάλλεται από εξωτερικό περικάλυμμα φαιάς ουσίας.

## 2.1.1 Η ανατομία του εγκεφάλου

Ο εγκέφαλος βρίσκεται μέσα στο κύτος του κρανίου και συνεχίζεται στο νωτιαίο μυελό μέσω του ινιακού τρήματος (*foramen magnum*) (Snell, 2010). Περιβάλλεται από τις τρεις μήνιγγες οι οποίες είναι η σκληρή μήνιγγα (*dura mater/ pachymeninx*), η αραχνοειδής μήνιγγα (*arachnoid mater*) και η χοριοειδής μήνιγγα (*pia mater*).



Εικόνα 1 Προστατευτικά περιβλήματα εγκεφάλου<sup>1</sup>

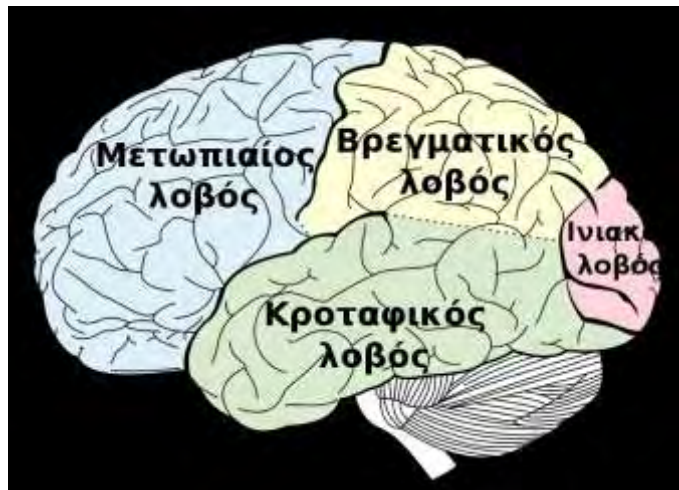
Ο εγκέφαλος υποδιαιρείται σε τρία μείζονα μέρη. Κατά ανιούσα διάταξη, από το νωτιαίο μυελό, τα μέρη αυτά είναι : ο ρομβοειδής ή οπίσθιος εγκέφαλος (*hindbrain/ rhombencephalon*) , ο μέσος εγκέφαλος (*midbrain*) και ο πρόσθιος εγκέφαλος (*procerebrum*). Ο ρομβοειδής εγκέφαλος υποδιαιρείται στο μυελεγκέφαλο(*myelencephalon*) ή προμήκη μυελό (*marrowbrain*), στο μετεγκέφαλο(*metencephalon*) ή γέφυρα(*pons*) , και στην παρεγκεφαλίδα(*cerebellum/parencephalis*). Ο πρόσθιος εγκέφαλος υποδιαιρείται στο διάμεσο εγκέφαλο (*interbrain*) και στον τελικό εγκέφαλο (*cerebrum=εγκέφαλος*).<sup>2</sup> Ο προμήκης μυελός, η γέφυρα και ο μεσεγκέφαλος αποτελούν το εγκεφαλικό στέλεχος. Το τμήμα αυτό του εγκεφάλου συνδέεται με το νωτιαίο μυελό και ελέγχει ακούσιες λειτουργίες του ανθρώπινου οργανισμού, όπως η συντονισμένη κίνηση της καρδιάς.

Ο οπίσθιος εγκέφαλος, όπως αναφέρεται και παραπάνω, υποδιαιρείται σε τρία μέρη: τον προμήκη μυελό, τη γέφυρα και την παρεγκεφαλίδα. Ο προμήκης μυελός συνδέει τη γέφυρα προς τα άνω, με το νωτιαίο μυελό προς τα κάτω. Περιέχει πολλές ομάδες νευρώνων και λειτουργεί ως αγωγός των νευρικών ινών(*nervefibres*). Η γέφυρα βρίσκεται μπροστά από την παρεγκεφαλίδα, κάτω από το μέσο εγκέφαλο και πάνω από τον προμήκη μυελό. Περιέχει, όπως και ο προμήκης μυελός, πολλούς πυρήνες(*core/kernel*) καθώς και νευρικές ίνες. Η παρεγκεφαλίδα, στη συνέχεια, βρίσκεται στον οπίσθιο κρανιακό βόθρο (*fossa*), πίσω από τη γέφυρα και τον προμήκη μυελό. Είναι το νευρικό όργανο που ρυθμίζει την ισορροπία και τον συντονισμό των κινήσεων των μυών, τόσο κατά την

<sup>1</sup> Η εικόνα αντλήθηκε από την πηγή (Snell, 2010)

κίνηση όσο και στην ακινησία. Ο μέσος εγκέφαλος συνδέει τον πρόσθιο και τον οπίσθιο εγκέφαλο. Επιπλέον περιέχει πολλούς πυρήνες και δεσμίδες ανερχόμενων και κατερχόμενων νευρικών ινών. Ο πρόσθιος εγκέφαλος υποδιαιρείται στο διάμεσο εγκέφαλο και στον τελικό εγκέφαλο. Ο διάμεσος εγκέφαλος είναι σχεδόν εξ ολοκλήρου κρυμμένος από την επιφάνεια του εγκεφάλου. Αποτελείται από μία ραχιαία μοίρα, το θάλαμο (*thalamus*) και μία κοιλιακή μοίρα, τον υποθάλαμο (*hypothalamus*). Όσον αφορά τον τελικό εγκέφαλο (*cerebrum*), το μεγαλύτερο μέρος του εγκεφάλου (*brain*), αποτελείται από τα δύο εγκεφαλικά ημισφαίρια (*hemisphere*), συνδεόμενα με μάζα λευκής ουσίας, το μεσολόβιο (*trabscerebri*). Τα ημισφαίρια είναι καθοριστικής σημασίας για τον εγκέφαλο καθώς είναι υπεύθυνα για τα ερεθίσματα που δέχεται και ερμηνεύει ο εγκέφαλος, για το υποσυνείδητο, καθώς και τη σωστή λειτουργία κατώτερων μελών όπως το στέλεχος και η παρεγκεφαλίδα.

Στον εγκεφαλικό φλοιό εμφανίζονται οι έλικες (*spiral*), οι οποίες χωρίζονται σε αύλακες ή σχισμές (*furrow*). Με αυτόν τον τρόπο αυξάνεται σημαντικά η έκταση του φλοιού. Μερικές από τις βαθύτερες αύλακες χρησιμοποιούνται κλασικά για τη συμβατική υποδιαίρεση της επιφάνειας κάθε ημισφαιρίου σε λοβούς. Όσον αφορά τους λοβούς, έχουμε τέσσερα είδη και παίρνουν το όνομα τους από το οστό υπό το οποίο βρίσκονται. Ο μετωπιαίος λοβός (*frontal lobe*) έχει ως ρόλο να ελέγχει συνειδητά το συναίσθημα και την έκφραση του λόγου, ο βρεγματικός λοβός (*parietal lobe*) ευθύνεται για τη λειτουργία των αισθήσεων όπως είναι η αφή, η πίεση και η θερμοκρασία, ο οπτικός προσανατολισμός και η ικανότητα προσανατολισμού. Επιπλέον, ο κροταφικός λοβός (*temporal lobe*) ελέγχει τη μνήμη και τις ειδικές αισθήσεις όπως είναι η ακοή καθώς επίσης και ικανότητες όπως είναι η κατανόηση γραπτών ή και προφορικών λέξεων. Τέλος, ο ινιακός λοβός (*occipital lobe*) ασκεί έλεγχο στην όραση του ατόμου.



Εικόνα 2 Οι λοβοί του εγκεφάλου (Gray, 1878)

## 2.1.2 Η νευροφυσιολογία του εγκεφάλου

Ο εγκέφαλος μαζί με το νωτιαίο μυελό και τα νεύρα αποτελούν τα όργανα του ανθρώπινου νευρικού συστήματος. Τα όργανα αυτά δομούνται από νευρικό ιστό και τα κύτταρα αυτού του είδους ιστού χωρίζονται σε δύο είδη: τα νευρικά κύτταρα (ή νευρώνες) και τα νευρογλοιακά κύτταρα.

Ένας τυπικός βιολογικός νευρώνας αποτελείται από το κυτταρικό σώμα(*body*) που αποτελεί τον πυρήνα του, τους δενδρίτες (*dendrites*) μέσω των οποίων λαμβάνει σήματα από γειτονικούς νευρώνες και το νευράξονα (ή νευρίτη)(*neuraxis* ή *neurite*), που είναι η έξοδος του νευρώνα και το μέσο σύνδεσής του με άλλους νευρώνες. Οι δενδρίτες κάθε νευρώνα συνδέονται με τους άξονες άλλων νευρώνων και τα σημεία αυτής της σύνδεσης λέγονται συνάψεις(*synapsis*). Οι συνάψεις μέσω χημικών διαδικασιών επιταχύνουν ή επιβραδύνουν τη ροή ηλεκτρικών φορτίων προς το σώμα του νευρώνα. Η ικανότητα μάθησης και μνήμης που παρουσιάζει ο εγκέφαλος οφείλεται στην ικανότητα των συνάψεων να μεταβάλλουν την αγωγιμότητά τους. Τα ηλεκτρικά σήματα που εισέρχονται στο σώμα των νευρώνων μέσω των δενδριτών συνδυάζονται κατάλληλα. Εάν το αποτέλεσμα ξεπεράσει μία τιμή κατωφλίου, το σήμα διαδίδεται με τη βοήθεια του άξονα προς άλλους νευρώνες. Οι νευρώνες παρουσιάζουν μορφολογικές και λειτουργικές διαφορές και διακρίνονται, ανάλογα με τη λειτουργία που επιτελούν, σε αισθητικούς, κινητικούς και ενδιάμεσους.

Τα νευρογλοιακά κύτταρα είναι πολύ περισσότερα από τους νευρώνες και έχουν βοηθητικό ρόλο. Έχουν ποικίλα σχήματα και ειδικές λειτουργίες. Προμηθεύουν με θρεπτικά συστατικά το νευρώνα και χρησιμεύουν στην απορρόφηση και απομάκρυνση των άχρηστων ουσιών από αυτούς. Τα νευρογλοιακά κύτταρα, που περιβάλλουν το νευράξονα των περισσότερων από τους νευρώνες, συμβάλλουν στη μόνωσή του και στην επιτάχυνση της μεταφοράς της νευρικής ώσης.

## 2.2 Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)(*Artificial Neural Networks(ANN)*) βασίζονται στη χρήση μίας προσέγγισης του τρόπου λειτουργίας του Κεντρικού Νευρικού Συστήματος, μέσω μαθηματικών συναρτήσεων και είναι συστήματα επεξεργασίας δεδομένων που αποτελούνται από ένα πλήθος τεχνητών νευρώνων οργανωμένων σε δομές παρόμοιες με αυτές του ανθρώπινου εγκεφάλου. Αποτελούν την πιο αντιπροσωπευτική κατηγορία συστημάτων που δημιουργήθηκαν εμπνευσμένα από τον ανθρώπινο εγκέφαλο. Οι τεχνητοί νευρώνες, συνήθως, είναι οργανωμένοι σε μία σειρά από επίπεδα ή στρώματα (*layers*), το πρώτο των οποίων ονομάζεται επίπεδο εισόδου (*input layer*) και χρησιμοποιείται για την εισαγωγή δεδομένων. Στη συνέχεια, ακολουθούν ένα ή περισσότερα ενδιάμεσα ή κρυφά επίπεδα (*hidden layers*), ενώ το τελικό επίπεδο είναι εκείνο της εξόδου (*output layer*) (Βλαχαβας, 2006).

### 2.2.1 Μοντέλο Τεχνητού Νευρώνα

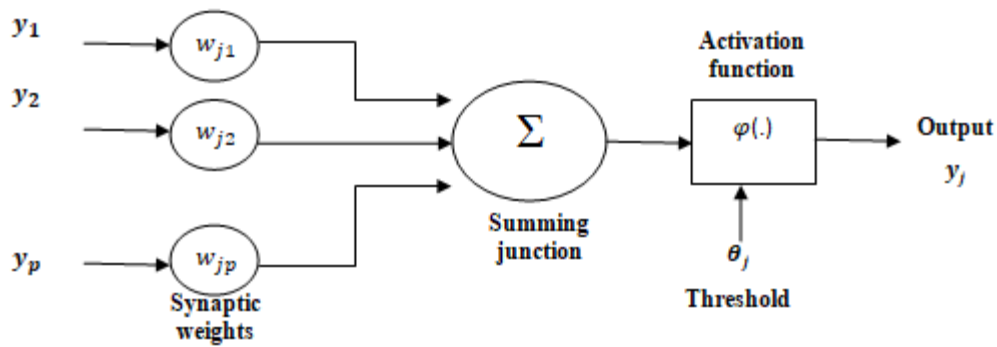
Υπάρχουν διάφορα είδη νευρώνων. Το είδος που θα επιλεγεί για να δομηθεί ένα συγκεκριμένο τεχνητό νευρωνικό δίκτυο, εξαρτάται από τη φύση του προβλήματος. Σε πολλές περιπτώσεις χρησιμοποιείται συνδυασμός διαφορετικών ειδών νευρώνα. Υπάρχουν οι βιολογικοί νευρώνες, οι οποίοι αναφέρονται σε συγκεκριμένα τμήματα νευρικού ιστού, όμως, υπάρχουν επίσης και οι τεχνητοί νευρώνες (*artificial neuron*) οι οποίοι αποτελούν αλγοριθμικό κατασκεύασμα. Στόχος τους μπορεί να

είναι η επίλυση κάποιου προβλήματος μαθηματικής - υπολογιστικής φύσεως ή ακόμα και η προσομοίωση λειτουργιών με χρήση συγκεκριμένου και γνωστού μαθηματικού μοντέλου.

Τα βασικά χαρακτηριστικά ενός τεχνητού νευρώνα είναι τα εξής:

- Τα συναπτικά βάρη (*synaptic weights*). Ένα σήμα  $x_j$  στην είσοδο ενός νευρώνα  $k$  σε μία σύναψη  $j$  πολλαπλασιάζεται με το συναπτικό βάρος  $W_{kj}$ . Ο πρώτος δείκτης αναφέρεται στο νευρώνα (στην περίπτωση αυτή ο νευρώνας  $k$ ) και ο δεύτερος στη σύναψη (στην περίπτωση αυτή η σύναψη  $j$ ), που δέχεται στην είσοδό της το σήμα  $W_{kj}$ . Το βάρος  $W_{kj}$  είναι θετικό, αν η σύναψη είναι διεγερτική και αρνητικό, αν η σύναψη είναι απαγορευτική. Το κάθε νευρωνικό δίκτυο αλληλεπιδρά με το περιβάλλον με διαφορετικό τρόπο και ρυθμό και η πληροφορία που αποκτά κωδικοποιείται στα συναπτικά βάρη. Το χαρακτηριστικό αυτό είναι που δίνει τη δυνατότητα της εξέλιξης και της προσαρμογής στο περιβάλλον. Τα βάρη μπορούν να πάρουν οποιαδήποτε τιμή, ωστόσο συνήθως κανονικοποιούνται.
- Ένας αθροιστής (*sum*) ο οποίος πραγματοποιεί την ολική πρόσθεση των σημάτων εισόδου που έχουν πολλαπλασιαστεί με τα αντίστοιχα συναπτικά βάρη.
- Η συνάρτηση ενεργοποίησης (*activation function*), η οποία περιορίζει και μειώνει το εύρος εξόδου ενός νευρώνα. Το κανονικοποιημένο πλάτος εξόδου του νευρώνα δίνει αποτέλεσμα στο διάστημα  $[0, 1]$  ή  $[-1, 1]$ , ανάλογα με τη συνάρτηση που επιλέχθηκε να υλοποιηθεί.
- Το κατώφλι/πόλωση  $\theta_k$  (*threshold, bias*). Πρόκειται για δευτερεύουσα παράμετρο του συστήματος, η οποία έχει επίδραση στην ελάττωση της εισόδου, στην εφαρμοζόμενη συνάρτηση ενεργοποίησης.

Όπως φαίνεται στην **Εικόνα 3**, το μοντέλο του νευρώνα αποτελείται κυρίως από τρεις φάσεις (Jain, Mao, & Mohiuddin, 1996; Kawaguchi, 2001a; Διαμανταρας, 2007). Το νευρωνικό δίκτυο όπως έχει ήδη αναφερθεί απαρτίζεται από νευρώνες στους οποίους η πληροφορία ρέει πάντα προς μία κατεύθυνση, από αριστερά προς τα δεξιά. Κατά την πρώτη φάση, λοιπόν, κάθε μία είσοδος πολλαπλασιάζεται με το αντίστοιχο συναπτικό βάρος. Στη συνέχεια, στη δεύτερη φάση, αφού οι είσοδοι έχουν σταθμιστεί, αθροίζονται και παράγεται το τοπικό πεδίο  $u_j$  (*net input, activation potential*). Στη συνέχεια, εφαρμόζεται η συνάρτηση ενεργοποίησης (*activation function*) στο τοπικό πεδίο και τελικά προκύπτει η έξοδος, η οποία είναι συνάρτηση της εισόδου. Για να είναι πιο αποτελεσματική η διαδικασία αυτή, περιλαμβάνεται και το κατώφλι  $\theta_j$  (*bias, threshold*).



Εικόνα 3 Σχηματική αναπαράσταση μη γραμμικού νευρώνα. Περιλαμβάνει τα συναπτικά βάρη (synaptic weights), έναν αθροιστή (summing junction), τη συνάρτηση ενεργοποίησης (activation function) και το κατώφλι (threshold)

Αν η διαδικασία σταματούσε στις δύο πρώτες φάσεις, τότε θα προέκυπτε ένας γραμμικός νευρώνας (*linear neuron*) και το νευρωνικό δίκτυο θα ήταν αντίστοιχα γραμμικό. Το πρώτο μοντέλο τέτοιας μορφής δημιουργήθηκε από τους Warren McCulloch and Walter Pitts, το 1943. Ειδικότερα, αποτελούνταν από τις εισόδους  $I_1, I_2, I_3, \dots, I_m$  και την έξοδο  $y$ , τα κανονικοποιημένα βάρη κυμαίνονταν στα διαστήματα  $[0, 1]$  ή  $[-1, 1]$ .

$$\sum I_i W_i \quad (2.1)$$

Ένας νευρώνας  $k$  περιγράφεται από τις σχέσεις:

$$u_k = \sum_{j=1}^q w_{kj} x_j \quad (2.2)$$

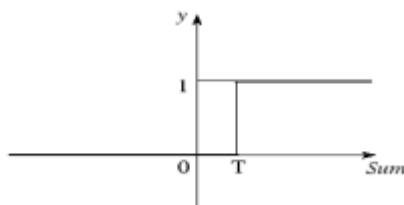
$$y_k = \varphi(u_k - \theta_k) \quad (2.3)$$

το κατώφλι  $\theta_k$  είναι εσωτερική παράμετρος του νευρώνα  $k$ . Έτσι μπορούμε να διαμορφώσουμε το συνδυασμό των εξισώσεων ως εξής:

$$u_k = \sum_{j=0}^q w_{kj} x_j \quad (2.4)$$

$$y_k = \varphi(u_k) \quad (2.5)$$

οι (2.4), (2.5) συμπεριλαμβάνουν τη σύναψη  $x_0 = -1$  και  $w_{k0} = \theta_k$



Εικόνα 4 Βηματική συνάρτηση για την τιμή κατωφλίου  $T$

Εάν το αποτέλεσμα είναι μικρότερο της τιμής κατωφλίου, τότε η έξοδος του νευρώνα είναι ίση με 0 και ο νευρώνας λέγεται αδρανής. Διαφορετικά, η έξοδος ισούται με 1 και ο νευρώνας ονομάζεται ενεργοποιημένος. Η συνάρτηση κατωφλίου είναι :

$$\varphi(u) = \begin{cases} 1, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (2.6)$$

και ως έξοδος του νευρώνα προκύπτει:

$$y_k = \begin{cases} 1, & u_k \geq 0 \\ 0, & u_k < 0 \end{cases} \quad (2.7)$$

Σήμερα, εκμεταλλευόμενοι τις ιδιότητες των παραγώγων και δεδομένου ότι η βηματική συνάρτηση είναι μη παραγωγίσιμη, χρησιμοποιείται, συνήθως, ως συνάρτηση ενεργοποίησης μία σιγμοειδής, η οποία είναι συνεχής, πραγματική και φραγμένη. Και σε αυτή την περίπτωση συνάρτησης, το σύνολο τιμών για τα συναπτικά βάρη είναι τα διαστήματα  $[0,1]$  ή  $[-1,1]$ .

Κάποιες σιγμοειδείς συναρτήσεις που χρησιμοποιούνται είναι η υπερβολική συνάρτηση (*hyperbolic function*), η συνάρτηση τόξου εφαπτομένης (*arctangent function*), και η υπερβολική εφαπτομένη (2.9). Η πιο συνηθισμένη, όμως, σιγμοειδής συνάρτηση, που χρησιμοποιείται ως συνάρτηση ενεργοποίησης, είναι η λογιστική συνάρτηση (*logistic function*), η οποία δίνεται από τον μαθηματικό τύπο (2.8):

$$S(x) = \frac{1}{1+e^{-\beta x}} \quad (2.8)$$

όπου  $\beta$  η παράμετρος κλίσης και όσο  $\beta \rightarrow \infty$  η συνάρτηση τείνει να γίνει βηματική.

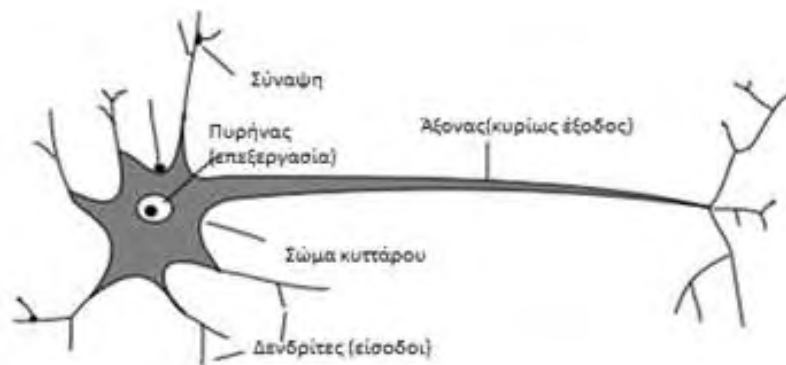
$$S(x) = \tanh \frac{\beta x}{2} \quad (2.9)$$

όπου  $\beta$  η παράμετρος κλίσης και όσο  $\beta \rightarrow \infty$  η συνάρτηση τείνει να γίνει συνάρτηση προσήμου.

Συνοπτικά, στους νευρώνες με:

- Βηματική συνάρτηση, η έξοδος μπορεί να είναι 0 ή 1,
- Σιγμοειδής συνάρτηση, η έξοδος μπορεί να είναι πραγματικός αριθμός στο  $[0,1]$ ,
- Συνάρτηση προσήμου, η έξοδος μπορεί να είναι -1 ή 1.

Στην **Εικόνα 5**, απεικονίζεται ο βιολογικός νευρώνας, στον οποίο έχει γίνει παραλληλισμός της λειτουργίας του με τα Τεχνητά Νευρωνικά Δίκτυα. Ειδικότερα, όπως φαίνεται και παρακάτω, οι δενδρίτες αντιστοιχούν στις εισόδους του ΤΝΔ, ο πυρήνας στην επεξεργασία του σήματος, ενώ ο άξονας στην κυρίως έξοδο.



**Εικόνα 5** Βιολογικός νευρώνας

## 2.2.2 Πλεονεκτήματα Νευρωνικών Δικτύων

Τα πλεονεκτήματα που επιφέρουν τα τεχνητά νευρωνικά δίκτυα είναι πολυάριθμα και παρουσιάζονται παρακάτω (Simon & Haykin, 2009).

- Μη γραμμικότητα (*Non linearity*): Ένας τεχνητός νευρώνας μπορεί να είναι γραμμικός ή μη γραμμικός. Ένα νευρωνικό δίκτυο, που αποτελείται από μια διασύνδεση των μη γραμμικών νευρώνων είναι μη γραμμικό.
- Χαρτογράφηση εισόδου-εξόδου (*Input-Output Mapping*): περιλαμβάνει την τροποποίηση των συναπτικών βαρών ενός νευρικού δικτύου. Το δίκτυο μαθαίνει με τον κατάλληλο σχεδιασμό εισόδου-εξόδου.
- Προσαρμοστικότητα (*Adaptivity*): τα νευρωνικά δίκτυα έχουν την ικανότητα να προσαρμόζουν τα συναπτικά βάρη τους ανάλογα με τις αλλαγές του περιβάλλοντος που βρίσκονται. Αν είναι εκπαιδευμένο, δηλαδή, να λειτουργεί σε ένα συγκεκριμένο περιβάλλον, μπορεί εύκολα να ανακατασκευαστεί για να ασχοληθεί με μικρές αλλαγές στις περιβαλλοντικές συνθήκες λειτουργίας.
- Αποδεικτική απόκριση (*Evidential Response*): ένα νευρωνικό δίκτυο μπορεί να σχεδιαστεί για να παρέχει πληροφορίες, όχι μόνο για το συγκεκριμένο υπόδειγμα που επιλέγεται, αλλά και για την εμπιστοσύνη στην απόφαση. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν ώστε να απορρίψουν τα ασαφή πρότυπα, εάν προκύψουν, και έτσι να βελτιώσουν την ταξινόμηση και τις επιδόσεις του νευρωνικού δικτύου.
- Συμφραζόμενη πληροφορία (*Contextual Information*): Η γνώση αντιπροσωπεύεται από την ίδια τη δομή και την κατάσταση ενεργοποίησης ενός νευρικού δικτύου. Κάθε νευρώνας επηρεάζεται ενδεχομένως από την δραστηριότητα όλων των υπόλοιπων νευρώνων στο δίκτυο.
- Αντοχή σε σφάλματα (*Fault Tolerance*): ένα νευρωνικό δίκτυο που είναι υλοποιημένο σε hardware έχει τη δυνατότητα να είναι ανθεκτικό σε σφάλματα.
- Δυνατότητα υλοποίησης σε VLSI (*Very-Large-Scale-Integrated (VLSI Implementability)*): Η παράλληλη φύση των νευρωνικών δικτύων δίνει τη δυνατότητα υλοποίησης σε πολύ μεγάλη κλίμακας τεχνολογία VLSI και αυτό έχει ως αποτέλεσμα να μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου με ιεραρχικό τρόπο.

## 2.2.3 Αλγόριθμος Ανάστροφης Μετάδοσης Σφάλματος

Μία από τις βασικές λειτουργίες των Τεχνητών Νευρωνικών Δικτύων είναι η εκπαίδευση (*training*) (Βλαχαβας, 2006) (Kawaguchi, 2001b). Η εκπαίδευση είναι η διαδικασία τροποποίησης της τιμής των βαρών των συνάψεων, ώστε δοθέντος συγκεκριμένου διανύσματος εισόδου, να παραχθεί συγκεκριμένο διάνυσμα εξόδου.



Η Ανάστροφη Μετάδοση Σφάλματος (*Back Propagation*) αποτελεί την πιο γνωστή μέθοδο εκπαίδευσης νευρωνικών δικτύων πολλών επιπέδων και στοχεύει στην ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος. Η μεταβολή των βαρών, σε αυτόν, βασίζεται στον υπολογισμό της συνεισφοράς κάθε βάρους στο συνολικό σφάλμα (Παπαμαρκος, 2015) (Rafael Gonzalez, 2008). Ο Αλγόριθμος Ανάστροφης Μετάδοσης Λάθους έχει χρησιμοποιηθεί σε εφαρμογές, όπως ο προσαρμοστικός έλεγχος των δυναμικών συστημάτων και σε πολλές γενικές εφαρμογές των νευρωνικών δικτύων. Σύμφωνα με αυτόν, το σφάλμα διαδίδεται προς τα πίσω (ανάστροφη διάδοση) και με τον τρόπο αυτό επιτυγχάνεται η κατάλληλη τροποποίηση των βαρών των συνάψεων. Για τη μείωση του σφάλματος, ο αλγόριθμος εξετάζει την κλίση της επιφάνειας σφάλματος και προσδιορίζει την κατεύθυνση της μεγαλύτερης κλίσης στην επιφάνεια σφάλματος. Η διεύθυνση αυτή χρησιμοποιείται για την ελαχιστοποίηση του σφάλματος. Ο αλγόριθμος επιλέγει την αρνητική κλίση, η οποία είναι η κατεύθυνση της απότομης καθόδου.

Αρχικά, η περιγραφή ξεκινάει από το επίπεδο εξόδου. Ο υπολογισμός του συνολικού τετραγωνικού σφάλματος ανάμεσα στις επιθυμητές αποκρίσεις  $r_q$  και στις αντίστοιχες πραγματικές  $O_q$  των κόμβων στο επίπεδο εξόδου  $Q$  δίνεται από τον τύπο:

$$E_q = \frac{1}{2} \sum_{q=1}^{N_Q} (r_q - O_q)^2 \quad (2.10)$$

όπου  $N_Q$  το πλήθος των κόμβων του επιπέδου εξόδου  $Q$ , ενώ ο συντελεστής  $1/2$  εξυπηρετεί στο να πραγματοποιηθούν, στη συνέχεια, ευκολότερα οι υπολογισμοί. Στόχος είναι η δημιουργία ενός κανόνα εκπαίδευσης που να επιτρέπει την προσαρμογή των βαρών σε κάθε στρώμα, με έναν τρόπο ο οποίος θα αναζητά ένα ελάχιστο μία συνάρτησης σφάλματος της μορφής (2.10). Η προσαρμογή των βαρών σε αναλογία με τη μερική παράγωγο του σφάλματος ως προς τα βάρη, οδηγεί στο τελικό αποτέλεσμα, το οποίο θα είναι:

$$\Delta w_{pq} = -\alpha \frac{\partial E_Q}{\partial w_{pq}} \quad (2.11)$$

όπου το επίπεδο  $P$  προηγείται του  $Q$  και ο συντελεστής  $\alpha$  αποτελεί μία θετική σταθερά διόρθωσης. Το σφάλμα  $E_Q$  αποτελεί μία συνάρτηση των εξόδων  $Q_q$ , οι οποίες είναι συναρτήσεις των εισόδων  $I_q$ . Χρησιμοποιώντας τη σύνθετη και μερική παράγωγο της συνάρτησης  $E_q$  προκύπτει ο τύπος:

$$\frac{\partial E_Q}{\partial w_{pq}} = \frac{\partial E_Q}{\partial I_q} \frac{\partial I_q}{\partial w_{pq}} \quad (2.12)$$

Η είσοδος στον κόμβο κάποιου στρώματος είναι ένα ζυγισμένο άθροισμα των εξόδων από το προηγούμενο στρώμα. Εάν συμβολιστεί ως  $K$  το επίπεδο που βρίσκεται πριν το επίπεδο  $J$ , τότε η είσοδος που στέλνεται στο στοιχείο επεξεργασίας του κάθε κόμβου του στρώματος  $J$ , η οποία συμβολίζεται ως  $I_j$  υπολογίζεται ως εξής:

$$I_j = \sum_{k=1}^{N_k} w_{jk} Q_k \quad (2.13)$$

για τις τιμές  $j = 1, 2, \dots, N_j$  όπου  $N_j$  είναι το πλήθος των κόμβων του στρώματος  $J$ ,  $N_k$  είναι το πλήθος των κόμβων του στρώματος  $K$  και  $w_{jk}$  είναι τα βάρη που τροποποιούν τις εξόδους  $O_k$  των κόμβων του

στρώματος  $K$  πριν αυτές διαβιβαστούν στους κόμβους του στρώματος  $J$ . Οι έξοδοι του στρώματος  $K$  είναι οι :

$$O_k = \square_k(I_k) \quad (2.14)$$

Η σχέση (2.12) με τη βοήθεια της σχέσης (2.13) γίνεται:

$$\frac{\partial I_q}{\partial w_{pq}} = \frac{\partial}{\partial w_{pq}} \sum_{p=1}^{N_p} w_{pq} O_p = O_p \quad (2.15)$$

Κάνοντας αντικατάσταση των εξισώσεων (2.13) και (2.14) στην (2.11) προκύπτει η σχέση:

$$\Delta w_{pq} = -\alpha \frac{\partial E_Q}{\partial w_{qp}} O_p = \alpha \delta_q O_p \quad (2.16)$$

όπου:

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} \quad (2.17)$$

Ο υπολογισμός της τιμής της μεταβλητής  $\delta_q$ , με τη βοήθεια τη σύνθετης παραγωγίσις γίνεται ως εξής:

$$\delta_q = -\frac{\partial E_Q}{\partial I_q} = -\frac{\partial E_Q}{\partial O_q} \frac{\partial O_q}{\partial I_q} \quad (2.18)$$

Με τη βοήθεια τη σχέσης (2.10), η σχέση (2.18) γίνεται:

$$\frac{\partial E_Q}{\partial O_q} = -(r_q - O_q) \quad (2.19)$$

Στη συνέχεια από τη σχέση (2.14) προκύπτει η νέα σχέση:

$$\frac{\partial O_q}{\partial I_q} = \frac{\partial}{\partial I_q} \square_q(I_q) = \square'_q(I_q) \quad (2.20)$$

Αντικαθιστώντας τις εξισώσεις (2.19) και (2.20) στην (2.18) θα προκύψει ο τύπος:

$$\delta_q = \left( (r_q - O_q) \square'_q(I_q) \right) \quad (2.21)$$

Με τη βοήθεια των τύπων (2.17)-(2.19) και (2.16) τελικά προκύπτει:

$$\Delta w_{qp} = \alpha (r_q - O_q) \square'_q(I_q) O_p = \alpha \delta_q O_p \quad (2.22)$$

Έπειτα από τον προσδιορισμό της συνάρτησης  $\square_q(I_q)$ , κατά την παρουσίαση οποιαδήποτε προτύπου εκπαίδευσης στην είσοδο του δικτύου, είναι γνωστή η επιθυμητή απόκριση  $r_q$  του κάθε κόμβου εξόδου.

Με παρόμοιο τρόπο και κατεύθυνση από το επίπεδο εξόδου προς τα πίσω, στη συνέχεια αναλύεται το επίπεδο  $P$ . Έτσι, εφαρμόζοντας την ίδια μεθοδολογία προκύπτει:

$$\Delta w_{pj} = \alpha (r_p - O_p) \square'_p(I_p) O_j = \alpha \delta_p O_j \quad (2.23)$$

όπου:

$$\delta_p = \left( (r_p - O_p) \square'_p(I_p) \right) \quad (2.24)$$

Λαμβάνοντας υπόψη τη σχέση (2.18) ο όρος του σφάλματος για το επίπεδο  $P$  ορίζεται ως εξής:

$$\delta_p = -\frac{\partial E_p}{\partial I_p} = -\frac{\partial E_p}{\partial O_p} \frac{\partial O_p}{\partial I_p} \quad (2.25)$$

όπου:

$$\frac{\partial O_p}{\partial I_p} = \frac{\partial}{\partial I_p} \square_p(I_p) = \square'_p(I_p) \quad (2.26)$$

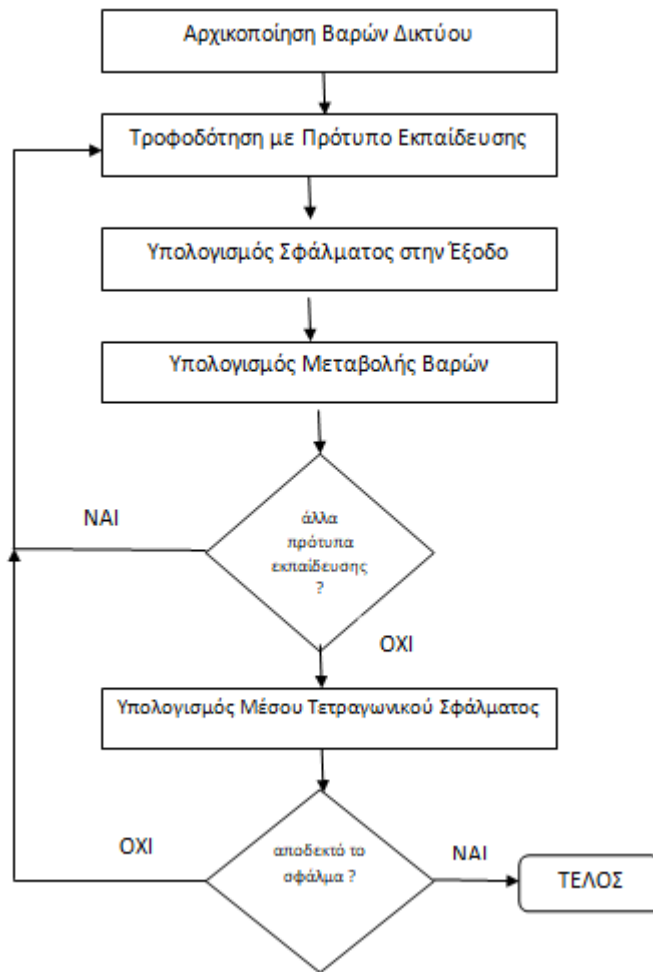
Με τη βοήθεια του κανόνα της σύνθετης παραγωγίσις προκύπτει:

$$-\frac{\partial E_p}{\partial O_p} = -\sum_{q=1}^{N_Q} \frac{\partial E_p}{\partial I_q} \frac{\partial I_q}{\partial O_p} = \sum_{q=1}^{N_Q} \left(-\frac{\partial E_p}{\partial I_q}\right) \frac{\partial}{\partial O_p} \sum_{p=1}^{N_p} w_{qp} O_p = \sum_{q=1}^{N_Q} \delta_q w_{qp} \quad (2.27)$$

Κάνοντας αντικατάσταση τις εξισώσεις (2.26) και (2.27) στην (2.25), τελικά το αποτέλεσμα θα είναι:

$$\delta_p = \square'_p(I_p) \sum_{q=1}^{N_Q} \delta_q w_{qp} \quad (2.28)$$

Δεδομένου, λοιπόν, πως όλες οι παράμετροι των σχέσεων (2.23) και (2.28) είναι γνωστοί, δίνεται η δυνατότητα να προσδιοριστεί πλήρως ένας κανόνας εκπαίδευσης για το επίπεδο  $P$ . Υπάρχουν πολλές τροποποιήσεις στο βασικό αλγόριθμο ανάστροφης διάδοσης με στόχο τη σύγκλιση του συστήματος. Ως σύγκλιση μπορεί να οριστεί μια μείωση του συνολικού σφάλματος, κάτω από ένα κατώτατο όριο και το σημείο εκείνο λέγεται πως είναι πλήρως εκπαιδευμένο.



Εικόνα 6 Διαδικασία Εκπαίδευσης Ανάστροφης Μετάδοσης Σφάλματος (Βλαχαβας, 2006)

## 2.2.4 Σύγχρονες εκδοχές νευρωνικών δικτύων

Τα Τεχνητά Νευρωνικά Δίκτυα χωρίζονται σε τέσσερις γενιές. Στην "1η Γενιά ΤΝΔ" ανήκουν κυρίως εφαρμογές στο Ν-διάστατο Ευκλείδειο χώρο,  $\square^N$  (Bishop, 2006; Βασιλειος Γ.Καμπουρλαζος, 2015). Αντιπροσωπευτικό παράδειγμα αυτής της κατηγορίας αποτελούν τα Perceptrons. Η "2η Γενιά ΤΝΔ" έχει ως βασικό περιεχόμενο ΤΝΔ με αρχιτεκτονικές βαθιάς μάθησης (*deep learning*). Χαρακτηριστικό παράδειγμα ΤΝΔ βαθιάς μάθησης αποτελούν τα ΤΝΔ βαθιάς συνέλιξης (*deep convolutional ANN*). Επιπρόσθετα, το 1979 ο Kunihiko Fukushima εισήγαγε τα πολυστρωματικά ΤΝΔ Neocognitron.

Το μοντέλο Neocognitron έχει δομή παρόμοια με το μοντέλο ιεραρχίας του οπτικού νευρικού συστήματος, δηλαδή του οπτικού φλοιού του ανθρώπινου εγκεφάλου, που πρότειναν οι Hubel και Wiesel (Fukushima, 1980). Οι νευρώνες του δικτύου είναι οργανωμένοι σε επίπεδα. Το επίπεδο εισόδου (*photo receptor array*), αποτελείται από μια διαδοχική σύνδεση ενός αριθμού δομημένων δομών με δύουπο-επίπεδα κυττάρων η καθεμία, τα οποία είναι συνδεδεμένα. Κάθε υποεπίπεδο διαιρείται σε έναν αριθμό στρωμάτων. Το πρώτο στρώμα αποτελείται από "S-κύτταρα", τα οποία εμφανίζουν χαρακτηριστικά παρόμοια με τα απλά κύτταρα ή τα υπερσύνθετα κύτταρα κατώτερης

τάξης και το δεύτερο στρώμα αποτελείται από "C-κύτταρα", τα οποία είναι παρόμοια με σύνθετα κύτταρα ή ανώτερα υπερσύνθετα κύτταρα. Οι συνάψεις σε κάθε S-κύτταρο είναι τροποποιήσιμες. Το δίκτυο αυτού του είδους έχει την ικανότητα να μαθαίνει χωρίς επίβλεψη. Επιπλέον, μία από τις εφαρμογές που έχουν χρησιμοποιηθεί τα Neocognitron TND είναι για την αναγνώριση χειρόγραφων χαρακτήρων.

Τα Ακιδωτά Τεχνητά Νευρωνικά Δίκτυα (*Spiking Neural Networks (SNNs)*) ανήκουν στην "3η Γενιά TND", έχοντας αυξημένο συγκριτικά με τις προηγούμενες γενιές, το επίπεδο ρεαλισμού, σε μία νευρική προσομοίωση (Ferster & Spruston, 1995) (Tavanaei, Ghodrati, Kheradpisheh, Masquelier, & Maida, 2018). Αυτό το πετυχαίνουν χρησιμοποιώντας μεμονωμένες αιχμές. Ως αποτέλεσμα επιτρέπεται η ενσωμάτωση επιπλέον των χωροχρονικών πληροφοριών στην επικοινωνία και τον υπολογισμό, όπως γίνεται και στους πραγματικούς νευρώνες. Ειδικότερα, αντί να χρησιμοποιούν κωδικοποίηση ρυθμού, χρησιμοποιούν κωδικοποίηση παλμών πετυχαίνοντας έτσι να επιτρέπεται η πολυπλεξία των πληροφοριών.

Πιο συγκεκριμένα, οι νευρώνες στα SNNs δεν πυροδοτούνται σε κάθε κύκλο διάδοσης (όπως συμβαίνει με τα τυπικά δίκτυα πολλαπλών στρώσεων (*perceptrons*)), αλλά μόνο όταν το δυναμικό μεμβράνης φτάσει μια συγκεκριμένη τιμή. Όταν ένα νεύρο πυροδοτείται, παράγει ένα σήμα το οποίο ταξιδεύει σε άλλους νευρώνες οι οποίοι, με τη σειρά τους, αυξάνουν ή μειώνουν τις δυνατότητές τους σύμφωνα με αυτό το σήμα. Στο πλαίσιο της δημιουργίας νευρωνικών δικτύων, το τρέχον επίπεδο ενεργοποίησης θεωρείται κανονικά ως κατάσταση του νευρώνα, με εισερχόμενες ακίδες που μεταβάλλουν αυτή την τιμή υψηλότερα και, στη συνέχεια, είτε πυροδοτούν, είτε αποσυντίθενται με την πάροδο του χρόνου.

Τέλος, η "4η Γενιά TND" ενδέχεται να εφαρμοστεί για υλοποιήσεις εκτός του χώρου  $\square^N$  (Βασιλειος Γ.Καμπουραζος, 2015).

## 2.2.5 Αλγόριθμοι εκπαίδευσης TND

Η εκπαίδευση είναι μία θεμελιακή ικανότητα των Τεχνικών Νευρωνικών Δικτύων, η οποία δίνει τη δυνατότητα να βελτιώνονται καθώς μαθαίνουν από το περιβάλλον τους (Conforth & Meng, 2008) (Sheta, Braik, & Al-Hiary, 2019). Ο αλγόριθμος εκπαίδευσης καθορίζει με συγκεκριμένο τρόπο κανόνες που πρέπει να ακολουθηθούν προκειμένου να μεταβληθούν κατάλληλα τα βάρη της συνάρτησης. Υπάρχει μεγάλη ποικιλία αλγορίθμων εκπαίδευσης, με διαφορετικά πλεονεκτήματα και μειονεκτήματα. Προκειμένου τα νευρωνικά δίκτυα να χρησιμοποιηθούν κατά το βέλτιστο και πιο αποδοτικό τρόπο, πολλές φορές κρίνεται ωφέλιμη η χρήση αλγορίθμων βελτιστοποίησης για την εκπαίδευσή τους.

Μία από τις κατηγορίες αλγορίθμων που χρησιμοποιούνται για εκπαίδευση των νευρωνικών δικτύων αποτελούν οι βιο-εμπνευσμένοι αλγόριθμοι οι οποίοι έχουν ως πηγή έμπνευσης τη βιολογία, και μελετώνται στην παρούσα πτυχιακή εργασία. Κάποιοι από αυτούς είναι:

<b>Βιο-εμπνευσμένος αλγόριθμος</b>	<b>Εφαρμογή σε Νευρωνικά Δίκτυα</b>
Γενετικός Αλγόριθμος (Genetic Algorithm (GA))	<ul style="list-style-type: none"> <li>• Ρύθμιση της δομής και των παραμέτρων Νευρωνικού Δικτύου χρησιμοποιώντας διακόπτες εισόδου-εξόδου(Leung, Lam, Ling, &amp; Tam, 2003)</li> </ul>
Αλγόριθμος Διαφορικής Εξέλιξης (Differential Evolution Algorithm (DE))	<ul style="list-style-type: none"> <li>• Εκπαίδευση feed-forward Νευρωνικού Δικτύου για τον προσδιορισμό μη γραμμικού συστήματος (Subudhi &amp; Jena, 2011)</li> </ul>
Αλγόριθμοι Βελτιστοποίησης Αποικιών Μυρμηγκιών (Ant Colony Optimization (ACO))	<ul style="list-style-type: none"> <li>• Εκπαίδευση feed-forward Νευρωνικού Δικτύου για ταξινόμηση προτύπων(Socha &amp; Blum, 2007)</li> </ul>
Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization (PSO))	<ul style="list-style-type: none"> <li>• Εκπαίδευση βαρών feed-forward Νευρωνικού Δικτύου σε συνδυασμό με αλγόριθμο αναστροφής μετάδοσης σφάλματος(J.-R. Zhang, Zhang, Lok, &amp; Lyu, 2007)</li> </ul>
Αλγόριθμος Τεχνητής Αποικίας Μελισσών (Artificial Bee Colony Optimization (ABC))	<ul style="list-style-type: none"> <li>• Εκπαίδευση βαρών feed-forward Νευρωνικού Δικτύου (Dervis Karaboga, Akay, &amp; Ozturk, 2007)</li> </ul>
Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών Brain Storm Optimization (BSO)	<ul style="list-style-type: none"> <li>• Ανάλυση και βελτιστοποίηση μονοδιάστατου φάσματος σε αβέβαια Νευρωνικά Δίκτυα (Xuejiao Ma, Jin, &amp; Dong, 2017)(D. Wang, Luo, Grunder, Lin, &amp; Guo, 2017)</li> </ul>
Αλγόριθμος Νυχτερίδας (Bat Algorithm (BA))	<ul style="list-style-type: none"> <li>• Τροποποιήσεις εξισώσεων ταχυτήτων, σε συνδυασμό με Ring και Master-Slave στρατηγικές (Jaddi, Abdullah, &amp; Hamdan, 2015)</li> </ul>
Αλγόριθμος Βελτιστοποίησης Σμήνους (Particle Swarm Optimization (PSO))	<ul style="list-style-type: none"> <li>• Εκπαίδευση βαρών feed-forward Νευρωνικού Δικτύου(Gudise &amp; Venayagamoorthy, 2003)</li> </ul>
Αλγόριθμος Ζευγαρώματος Πουλιού (Bird Mating Algorithm (BMA))	<ul style="list-style-type: none"> <li>• Εκπαίδευση βαρών Τεχνητών Νευρωνικών Δικτύων (Askarzadeh &amp; Rezazadeh, 2013)</li> </ul>
Αλγόριθμος Σμήνους Γατών (Cat Swarm Optimization Algorithm (CSA))	<ul style="list-style-type: none"> <li>• Βελτιστοποίηση συνδέσεων βαρών Τεχνητών Νευρωνικών Δικτύων (Yusiong, 2012)</li> </ul>
Αλγόριθμος της Μύγας των Φρούτων (Fruit Fly Optimization Algorithm (FOA))	<ul style="list-style-type: none"> <li>• Εύρεση κατάλληλης τιμής παραμέτρου εξάπλωσης μοντέλου πρόβλεψης φόρτισης ισχύος σε Νευρωνικά Δίκτυα παλινδρόμησης(H.-Z. Li, Guo, Li, &amp; Sun, 2013)</li> </ul>
Αλγόριθμος Βελτιστοποίησης Διθητικών	<ul style="list-style-type: none"> <li>• Εκπαίδευση Νευρωνικών Δικτύων για σχεδίαση</li> </ul>

Ζιζανίων (Invasive Weed Optimization Algorithm (IWO))	ενός νέου διαμορφωτή δέσμης κεραίας (Zaharis, Skeberis, Xenos, Lazaridis, & Cosmas, 2013)
Αλγόριθμος Επιλογής Κλώνων (Clonal Selection Algorithm (CSA))	<ul style="list-style-type: none"> <li>• Εκπαίδευση Νευρωνικών Δικτύων Κυματιδίου για πρόγνωση αιολικής ενέργειας (Chitsaz, Amjady, &amp; Zareipour, 2015)</li> </ul>
Μέθοδος Αρκτικής Γαρίδας (Krill Herd (KH))	<ul style="list-style-type: none"> <li>• Εκπαίδευση βαρών Τεχνητών Νευρωνικών Δικτύων (Kowalski &amp; Lukasik, 2016)</li> </ul>
Αναζήτησης Τροφής Βακτηρίων (Bacterial Foraging Optimization Algorithm (BFO))	<ul style="list-style-type: none"> <li>• Εκπαίδευση Νευρωνικών Δικτύων Κυματιδίων μέσω στάθμισης των παραμέτρων μετάφρασης και διαστολής των κόμβων του κυματιδίου (Ulagammai, Venkatesh, Kannan, &amp; Padhy, 2007)</li> </ul>

### 3.Βιο-εμπνευσμένοι αλγόριθμοι

Υπάρχουν πολλά προβλήματα τόσο στο χώρο των επιστημών όσο και στην πραγματική ζωή. Στόχος είναι να βρεθεί η καλύτερη δυνατή λύση για το εκάστοτε είδος προβλήματος και σε αυτό βοηθάει η σωστή τυποποίηση της περιγραφής του. Όπως φαίνεται και από τις βιβλιογραφικές αναζητήσεις, υπάρχουν αμέτρητοι αλγόριθμοι και πολλές φορές συγκεκριμένοι προβληματισμοί, συμπεριφορές και καταστάσεις δίνουν αφορμή για να σχεδιαστούν νέοι. Οι αλγόριθμοι, κατά την πλειονότητα, διαφέρουν μεταξύ τους σε αρκετές ιδιότητες, όπως για παράδειγμα, στην πολυπλοκότητα, στο χρόνο υλοποίησης, στη μνήμη και στην αποδοτικότητα.

Η πηγή έμπνευσης για τη δημιουργία ενός αλγορίθμου μπορεί να προέρχεται από πολλούς τομείς. Μία μεγάλη πηγή έμπνευσης αποτελεί η φύση, καθώς τα τελευταία έτη πολλοί αλγόριθμοι έχουν δημιουργηθεί από ζωικούς και φυτικούς οργανισμούς καθώς και φυσικά φαινόμενα. Έτσι, όπως αναφέρουν και οι (FisterJr, Yang, Fister, Brest, & Fister, 2013a) η πλειοψηφία των αλγορίθμων που είναι εμπνευσμένοι από τη φύση (*nature-inspired algorithms*) βασίζονται σε επιτυχημένα χαρακτηριστικά του βιολογικού συστήματος και έχουν ονομαστεί βιο-εμπνευσμένοι αλγόριθμοι (*bio-inspired algorithms*).

#### 3.2 Μεθοδολογία Βιβλιογραφικής Αναζήτησης

Στην ενότητα αυτή παρουσιάζονται, έπειτα από εκτενή βιβλιογραφική αναζήτηση, οι ομαδοποιήσεις βασικών βιο-εμπνευσμένων αλγορίθμων, καθώς επίσης και κάποια χαρακτηριστικά και εφαρμογές σε συγκεκριμένους τομείς. Η έρευνα έχει ως στόχο την καλύτερη κατανόηση, ως προς τη χρήση και το περιεχόμενο των αλγορίθμων, μέσα από τη συνοπτική αλλά περιεκτική περιγραφή αυτών, με τη βοήθεια πινάκων και στατιστικών δεδομένων.

Η μεθοδολογία της βιβλιογραφικής αναζήτησης, για να γίνει πιο αποτελεσματική, χωρίστηκε σε επιμέρους φάσεις. Κατά την πρώτη φάση, διεκπεραιώνεται η αναζήτηση των αλγορίθμων από δημοσιεύσεις, στις βιβλιογραφικές βάσεις δεδομένων Scopus και Google Scholar. Αφού συγκεντρώνεται το απαιτούμενο υλικό, κρίνεται ωφέλιμη η δημιουργία συγκεντρωτικών πινάκων και διαγραμμάτων. Πιο συγκεκριμένα, καταγράφονται όλα τα άρθρα που περιλαμβάνουν στον τίτλο ή το περιεχόμενο τους, τον αλγόριθμο για τον οποίο γίνεται η αναζήτηση. Επιπρόσθετα, δίνεται περισσότερη έμφαση της αναζήτησης στο έτος 2017.

Στη δεύτερη φάση, πραγματοποιείται η αναγνώριση και η ένταξη των αλγορίθμων σε κατηγορίες, ενώ επίσης ερευνούνται και καταγράφονται οι πιο πρόσφατες εφαρμογές στον τομέα, κυρίως, των βιοϊατρικών επιστημών.

Τέλος, κατά την τρίτη φάση, εντοπίζεται ο πιο κατάλληλος αλγόριθμος, ο οποίος είναι πρόσφατος, εμπνευσμένος από τον εγκέφαλο και έχει αρκετές προοπτικές βελτίωσης και χρήσης σε ποικίλες εφαρμογές. Έτσι, περιγράφεται και δοκιμάζεται πειραματικά, ώστε να διεξαχθούν τα αντίστοιχα συμπεράσματα.



### 3.3 Αποτελέσματα Βιβλιογραφικής Αναζήτησης

Η βιολογία αποτελεί σημαντική πηγή έμπνευσης για τη δημιουργία αλγορίθμων γι' αυτό και υπάρχουν πολυάριθμα είδη, που δημιουργήθηκαν με βάση τις φυσιολογικές λειτουργίες των φυτικών και ζωικών οργανισμών. Παρακάτω αναφέρεται ονομαστικά ένα μεγάλο μέρος βιο-εμπνευσμένων αλγορίθμων, κάποιοι από τους οποίους, σε επόμενη υποενότητα, περιγράφονται αναλυτικότερα, όσον αφορά τον τρόπο λειτουργίας τους καθώς και εφαρμογές που έχουν χρησιμοποιηθεί. Ειδικότερα, οι αλγόριθμοι αυτοί είναι:

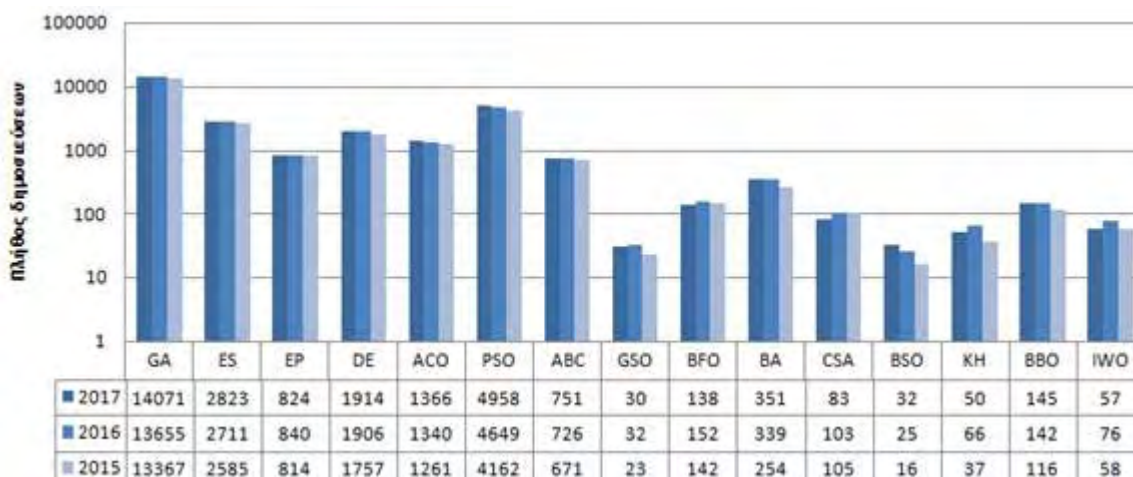
- Γενετικός Αλγόριθμος (Genetic Algorithm (GA))(Holland, 1984),
- Εξελικτικός Προγραμματισμός (Evolutionary Programming (EP))(D. B. Fogel & Fogel, 1995)
- Αλγόριθμος Διαφορικής Εξέλιξης (Differential Evolution Algorithm (DE))(Back, 1996)
- Αλγόριθμοι Βελτιστοποίησης Αποικιών των Μυρμηγκιών (Ant Colony Optimization (ACO)) (Dorigo & Di Caro, 1999),
- Αλγόριθμος Σμήνους Ακρίδων (Locust Swarm Algorithm (LSA))(S. Chen, 2009),
- Αλγόριθμος Μέλισσας Βομβίνου (Bumble Bee Mating Optimization Algorithm (BBMO)) (Marinakis & Marinaki, 2014)
- Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization (PSO)) (Eberhart & Kennedy, 1995)
- Αλγόριθμος Λαμπυρίδων (Firefly Algorithm (FA)) (X.-S. Yang, 2009),
- Αλγόριθμος Γκριζού Λύκου (Grey Wolf Algorithm (GWA))(Mirjalili, Mirjalili, & Lewis, 2014)
- Αλγόριθμος Βελτιστοποίησης Αναζήτησης Πιγκουίνων (Penguins Search Optimization Algorithm (PeSOA)) (Gheraibia & Moussaoui, 2013)
- Αλγόριθμος Αυτοκρατορικού Πιγκουίνου (Emperor Penguin Optimizer (EPO)) (Dhiman & Kumar, 2018)
- Αλγόριθμος Αράχνης (Social Spider Algorithm (SSA)) (Cuevas, Cienfuegos, Zaldivar, & Pérez-Cisneros, 2013)
- Αλγόριθμος Μετανάστευσης Ζώων (Animal Migration Optimization Algorithm (AMO)) (X. Li, Zhang, & Yin, 2014)
- Αλγόριθμος Φάλαινας (Whale Algorithm (WA)) (Mirjalili & Lewis, 2016)
- Αλγόριθμος Σπέρματος Φάλαινας (Sperm Whale Algorithm (SWA)) (Ebrahimi & Khamnehchi, 2016)
- Αλγόριθμος Τεχνητής Αποικίας Μελισσών (Artificial Bee Colony Optimization (ABC)) (Dervis Karaboga, Gorkemli, Ozturk, & Karaboga, 2014)
- Αλγόριθμος της Μύγας των Φρούτων (Fruit Fly Optimization Algorithm (FOA))(Pan, 2012)
- Αλγόριθμος βελτιστοποίησης Σουπιάς (Cuttlefish Algorithm (CA)) (Eesa, Brifcani, & Orman, 2013)

- Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων (Glowworm Swarm based Optimization (GSO)) (Krishnanand & Ghose, 2009)
- Αλγόριθμος Μαϊμούδων (Monkey Search Algorithm (MSA)) (Y. Zhou, Chen, & Zhou, 2016)
- Μέθοδος Αναζήτησης Τροφής Βακτηρίων (Bacterial Foraging Optimization Algorithm (BFO)) (Das, Biswas, Dasgupta, & Abraham, 2009)
- Αλγόριθμος Αποικίας Μελισσών (OptBees Algorithm) (Maia, Castro, & Caminhas, 2012)
- Αλγόριθμος Πεταλούδας Μονάρχη (Monarch Butterfly Algorithm (MBA)) (G.-G. Wang, Deb, & Cui, 2019)
- Αλγόριθμος Νυχτοπεταλούδας (Moth-Flame Algorithm (MFA)) (Mirjalili, 2015)
- Αλγόριθμος Σμήνους Ψαριών (Fish Swarm based Algorithm (FSA)) (Tsai & Lin, 2011)
- Αλγόριθμος Βελτιστοποίησης Λιονταριού (Lion Optimization Algorithm (LOA)) (Yazdani & Jolai, 2016)
- Αλγόριθμος Βελτιστοποίησης Κούκου (Cuckoo Search Algorithm (CSA)) (A. H. Gandomi, Yang, & Alavi, 2013)
- Αλγόριθμος Βελτιστοποίησης Σμήνους Δελφινιών (Dolphin Swarm Algorithm (DSA)) (T. Wu, Yao, & Yang, 2016)
- Αλγόριθμος Βελτιστοποίησης Όσφρησης Καρχαρία (Shark Smell Optimization Algorithm (SSO)) (Gnanasekaran, Chandramohan, Kumar, & Imran, 2016)
- Αλγόριθμος Σμήνους Γατών (Cat Swarm Optimization Algorithm (CSO)) (S.-C. Chu, Tsai, & Pan, 2006)
- Αλγόριθμος Νυχτερίδας (Bat Algorithm (BA)) (A. H. Gandomi, Yang, Alavi, & Talatahari, 2013)
- Αλγόριθμος Επιλογής Κλώνων (Clonal Selection Algorithm (CSA)) (Cutello, Narzisi, Nicosia, & Pavone, 2005)
- Αλγόριθμος Βελτιστοποίησης Καταγισμού Ιδεών (Brain Storm Optimization (BSO)) (Shi, 2011)
- Μέθοδος Αρκτικής Γαρίδας (Krill Herd (KH)) (A. Gandomi & Alavi, 2012)
- Αλγόριθμος Βελτιστοποίησης βασισμένος στη Βιογεωγραφία (Biogeography-based Optimization Algorithm (BBO)) (Simon, 2008)
- Αλγόριθμος Βελτιστοποίησης Δημητικών Ζιζανίων (Invasive Weed Optimization Algorithm (IWO)) (Mehrabian & Lucas, 2006)
- Αλγόριθμος Ζευγαρώματος Πουλίου (Bird Mating Algorithm (BMA)) (Askarzadeh, 2014)
- Αλγόριθμος Βελτιστοποίησης Γρύλλου (Cricket Algorithm (CA)) (Canayaz & Karci, 2016).

Στην **Εικόνα 7** παρουσιάζεται το πλήθος των δημοσιεύσεων που βρίσκεται καταχωρημένο στις βάσεις δεδομένων Scopus και Google Scholar, σε έρευνα που έγινε το Δεκέμβριο του 2018, συγκεκριμένων αλγορίθμων, κατά τη διάρκεια των ετών 2015-2017. Επιλέχθηκαν ενδεικτικά κάποιοι

αλγόριθμοι για στατιστική ανάλυση δεδομένων και εφαρμογών, οι οποίοι, είτε υπάρχουν σε μεγάλο βαθμό σε πολλές δημοσιεύσεις, είτε δεν έχουν χρησιμοποιηθεί ακόμα τόσο. Επίσης, προτιμήθηκε να καταγραφούν αλγόριθμοι που ανήκουν σε διαφορετικές κατηγορίες βιο-εμπνευσμένων αλγορίθμων. Οι κατηγορίες αυτές παρουσιάζονται, στη συνέχεια, στην **Εικόνα 8**, της επόμενης υποενότητας. Ο κάθετος άξονας είναι αριθμημένος σε λογαριθμική κλίμακα και αντιπροσωπεύει το πλήθος των δημοσιεύσεων, ενώ ο οριζόντιος άξονας αντιπροσωπεύει τα είδη των αλγορίθμων που εξετάζονται. Όπως φαίνεται και στην παρακάτω εικόνα, ο κάθε αλγόριθμος αποτελείται από τρεις ράβδους χρωματισμένες σε διαφορετικές αποχρώσεις του μπλε ανάλογα με το έτος. Πιο συγκεκριμένα, το μπλε σκούρο αναφέρεται στο 2017, το μπλε ανοιχτό στο 2016 και το γαλάζιο στο 2015. Οι αλγόριθμοι που περιγράφονται στο σχήμα είναι οι παρακάτω: Γενετικός Αλγόριθμος (Genetic Algorithm (GA)), Εξελικτικές Στρατηγικές (Evolutionary Strategy (ES)), Εξελικτικός Προγραμματισμός (Evolutionary Programming (EP)), Αλγόριθμος Διαφορικής Εξέλιξης (Differential Evolution Algorithm (DE)), Αλγόριθμοι Βελτιστοποίησης Αποικιών των Μυρμηγκιών (Ant Colony Optimization (ACO)), Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization (PSO)), Αλγόριθμος Τεχνητής Αποικίας Μελισσών (Artificial Bee Colony Optimization (ABC)), Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων (Glowworm Swarm based Optimization (GSO)), Μέθοδος Αναζήτησης Τροφής Βακτηρίων (Bacterial Foraging Optimization Algorithm (BFO)), Αλγόριθμος Νυχτερίδας (Bat Algorithm (BA)), Αλγόριθμος Επιλογής Κλώνων (Clonal Selection Algorithm (CSA)), Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών (Brain Storm Optimization (BSO)), Μέθοδος Αρκτικής Γαρίδας (Krill Herd (KH)), Αλγόριθμος Βελτιστοποίησης βασισμένος στη Βιογεωγραφία (Biogeography-based Optimization Algorithm (BBO)), Αλγόριθμος Βελτιστοποίησης Δηθητικών Ζιζανίων (Invasive Weed Optimization Algorithm (IWO)).

Συμπερασματικά, στην πλειοψηφία των αλγορίθμων παρατηρείται αύξηση του πλήθους δημοσιεύσεων και κατά συνέπεια αύξηση της χρήσης των αλγορίθμων μεταξύ των ετών 2015-2017. Έτσι, βρέθηκαν 27.592 δημοσιεύσεις το 2017, 26.762 δημοσιεύσεις το 2016, 25.368 δημοσιεύσεις το 2015. Συνολικά, κατά τη διάρκεια των ετών 2015-2017, οι δημοσιεύσεις ανέρχονται στις 79.722. Όπως είναι φανερό, υπάρχει αύξηση της χρήσης όλων των αλγορίθμων το έτος 2017 συγκριτικά με το 2015. Το 2017 περισσότερες εφαρμογές γίνονται στους Γενετικούς Αλγορίθμους (14.071), ενώ λιγότερο χρησιμοποιείται ο Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων (30).

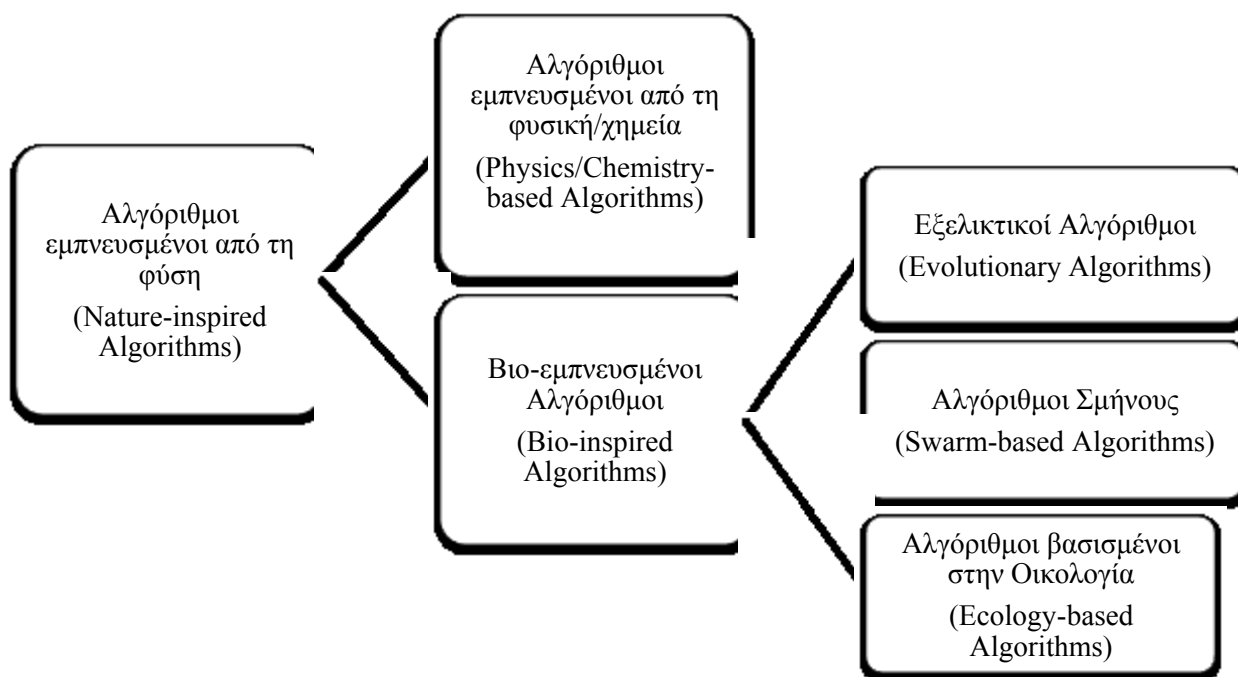


Εικόνα 7 Πλήθος δημοσιεύσεων ανά κατηγορία βιο-εμπνευσμένου αλγορίθμου στις βάσεις δεδομένων Scopus, Google Scholar

### 3.4 Καταγραφή και Ομαδοποίηση Αλγορίθμων

Υπάρχουν πολλοί αλγόριθμοι οι οποίοι διαφέρουν ως προς τη λειτουργία και τα χαρακτηριστικά τους. Κρίνεται ωφέλιμο, λοιπόν, να πραγματοποιηθεί η κατηγοριοποίηση καθώς και η ταξινόμησή τους, διαδικασίες που δεν είναι πάντοτε εύκολες και εφικτές. Είναι σημαντικό να αναφερθεί πως δεν υπάρχει μόνο ένας τρόπος ταξινόμησης των αλγορίθμων. Το πώς θα ενταχθούν στις επιμέρους κατηγορίες, όπως παρουσιάζεται και στη συνέχεια, **Εικόνα 8**, εξαρτάται από την οπτική προσέγγιση του χρήστη, για παράδειγμα μπορεί να υπάρξει διαφορετική διάταξη των αλγορίθμων ανάλογα το σκοπό χρήσης ή την πηγή έμπνευσης.

Η πηγή της έμπνευσης για τη δημιουργία του εκάστοτε αλγορίθμου, όπως αναφέρθηκε και προηγουμένως, ποικίλει σε πολύ μεγάλο βαθμό. Στην εργασία αυτή, όπως φαίνεται και στους παρακάτω πίνακες, οι αλγόριθμοι που είναι εμπνευσμένοι από τη φύση (*nature-inspired algorithms*), οι οποίοι βασίζονται σε χαρακτηριστικά που έχουν αντλήσει από βιολογικά συστήματα, ταξινομούνται σε δύο κύριες κατηγορίες, τους αλγορίθμους που είναι εμπνευσμένοι από τη φυσική/χημεία (*physics/chemistry-based algorithms*) και τους βιο-εμπνευσμένους αλγορίθμους (*biology-inspired/bio-inspired algorithms*), οι οποίοι υπερτερούν σε αριθμό και χωρίζονται σε υποκατηγορίες. Λαμβάνοντας υπόψη τις ομαδοποιήσεις των (Fister Jr, Yang, Fister, Brest, & Fister, 2013b), (Goel, Sharma, & Panchal, 2014), (Dhiman & Kumar, 2017), (G. Li, Jin, Akram, Chen, & Ji, 2018) και (Binitha, Sathya, & others, 2012a) προκύπτουν οι διαχωρισμοί που παρουσιάζονται στους ακόλουθους πίνακες σε κατηγορίες και είδη.



Εικόνα 8 Ομαδοποίηση βιο-εμπνευσμένων αλγορίθμων

Πίνακας 1 Κατηγορίες αλγορίθμων

Κατηγορία	Είδος	Εφαρμογές
<b>Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms)</b>	Γενετικοί Αλγόριθμοι Genetic Algorithms (GA)	<ul style="list-style-type: none"> <li>Μέθοδος πρόβλεψης transposon-derived piRNAs (D. Li, Luo, Zhang, Liu, &amp; Luo, 2016)</li> <li>Ανάλυση δεδομένων συστήματος ηλεκτροεγκεφαλογραφήματος (Electroencephalography (EEG))(Mehrnam, Nasrabadi, Ghodousi, Mohammadian, &amp; Torabi, 2017)</li> </ul>
	Εξελικτικές Στρατηγικές Evolutionary Strategy (ES)	<ul style="list-style-type: none"> <li>Βελτιωμένη παραγωγή αιθανόλης μέσω του βακτηρίου <i>Geobacillus thermoglucosidasius</i> (J. Zhou, Wu, &amp; Rao, 2016)</li> <li>Βελτιστοποίηση ραδιοσυχνότητας σε ασύρματα επαναφορτιζόμενα δίκτυα αισθητήρων (Yao, Jiang, Tsai, &amp; Wu, 2017)</li> </ul>
	Εξελικτικός Προγραμματισμός Evolutionary Programming (EP)	<ul style="list-style-type: none"> <li>Θεωρητική αξιολόγηση των παραμέτρων του μεταβολισμού της γλυκόζης με μαθηματική μοντελοποίηση (Sveshnikova et al., 2017)</li> </ul>
	Αλγόριθμος Διαφορικής Εξέλιξης Differential Evolution	<ul style="list-style-type: none"> <li>Υπολογισμός επιπέδων γλυκόζης στο αίμα (Koutny, 2016)</li> <li>Πλαίσιο εντοπισμού μετάλλαξης και</li> </ul>

	Algorithm (DE)	εξέλιξης καρκίνου (W. Zhang & Wang, 2017)
	Αλγόριθμοι Γενετικού Προγραμματισμού Genetic Programming Algorithms (GPA)	<ul style="list-style-type: none"> <li>• Πρόβλεψη δευτεροταγούς δομής πρωτεΐνης με νέες κωδικοποιήσεις.(Zamani &amp; Kremer, 2016)</li> <li>• Ταξινόμηση fMRI εικόνων καπνιστών πριν και μετά από θεραπεία (Tahmassebi et al., 2017)</li> </ul>
<p style="text-align: center;"><b>Αλγόριθμοι Σμήνους</b> <b>(Swarm-based Algorithms)</b> (Κοινωνικό Φαινόμενο-Social Phenomenon, Ανθρώπινο Ανοσοποιητικό Σύστημα-Human Immune System, Βιολογικό Ενδοκρινές Σύστημα-Biological Endocrine System)</p>	Αλγόριθμοι Βελτιστοποίησης Αποικιών των Μυρμηγκιών Ant Colony Optimization (ACO)	<ul style="list-style-type: none"> <li>• Ανίχνευση αιμοφόρων αγγείων σε ψηφιακές εικόνες αμφιβληστροειδούς(Bajeta, Sekuli, Djukanovi, Popovic, &amp; PoponiBugarin, 2016)</li> <li>• Μελέτη διασυνδέσεων εγκεφάλου από δεδομένα fMRI(Liu, Ji, Zhang, &amp; Liang, 2016)</li> </ul>
	Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων Particle Swarm Optimization Algorithm (PSOA)	<ul style="list-style-type: none"> <li>• Ομαδοποίηση δεδομένων Ηλεκτροκαρδιογραφήματος (electrocardiogram (ECG /EKG) ) καρδιακής αρρυθμίας (Raj, Ray, &amp; Shankar, 2016)</li> <li>• Εκτίμηση ισομορφών πρωτεϊνών(Zhao &amp; Feng, 2016)</li> </ul>
	Αλγόριθμος Τεχνητής Αποικίας Μελισσών Artificial Bee Colony Optimization Algorithm(ABC)	<ul style="list-style-type: none"> <li>• Ταξινόμηση μικροσυστοιχιών DNA (Garro, Rodriguez, &amp; Vázquez, 2016)</li> <li>• Ανίχνευση περιοχών πρόσδεσης μεταγραφικού παράγοντα σε ακολουθία DNA (D Karaboga &amp; Aslan, 2016)</li> </ul>
	Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων Glowworm Swarm based Optimization algorithm (GSO)	<ul style="list-style-type: none"> <li>• Αναγνώριση σύνθετων πρωτεϊνών σε δίκτυα πρωτεϊνικών αλληλεπιδράσεων(Lei, Li, Zhang, &amp; Wu, 2017)</li> </ul>
	Μέθοδος Αναζήτησης Τροφής Βακτηρίων Bacterial Foraging Optimization Algorithm (BFOA)	<ul style="list-style-type: none"> <li>• Εντοπισμός για εξάλειψη καρκινικών κυττάρων με χρήση νανορομποτ (J. Cao, Li, Wang, Huang, &amp; Zhao, 2016)</li> <li>• Πρόβλεψη δομής πρωτεϊνών (Ramayachitra &amp; Veeralakshmi, 2017)</li> </ul>

	Αλγόριθμος Νυχτερίδας Bat Algorithm (BA)	<ul style="list-style-type: none"> <li>Ταξινόμηση ιατρικών δεδομένων για διάγνωση Parkinson(Anci &amp; Dogantekin, 2016)</li> </ul>
	Αλγόριθμος Επιλογής Κλώνων Clonal Selection Algorithm (CSA)	<ul style="list-style-type: none"> <li>Μηχανές Διανυσμάτων Στήριξης για λανθασμένη διάγνωση καταστάσεων βιομηχανικών συστημάτων (Oliveira Batista, Rodrigues, &amp; Varejão, 2017)</li> <li>Μελέτη σημάτων Ηλεκτροεγκεφαλογραφήματος (ElMajdoui, Bougrine, Rbough, &amp; ElImrani, 2017)</li> </ul>
	Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών Brain Storm Optimization (BSO)	Περιγράφεται αναλυτικά στη συνέχεια
	Μέθοδος Αρκτικής Γαρίδας Krill Herd (KH)	<ul style="list-style-type: none"> <li>Αξιολόγηση γλαυκώματος (Sambandam, Josephine, &amp; Jeyabalaraja, 2017)</li> </ul>
<b>Οικολογία(Ecology)</b>	Αλγόριθμος Βελτιστοποίησης βασισμένος στη Βιογεωγραφία Biogeography-based Optimization Algorithm (BBO)	<ul style="list-style-type: none"> <li>Σχεδιασμός νέων σπειρών (coils) για τη μαγνητική διέγερση των νευρών (Di Barba, Mognaschi, &amp; Krawczyk, 2017)</li> </ul>
	Αλγόριθμος Βελτιστοποίησης Δηθητικών Ζιζανίων Invasive Weed Optimization Algorithm (IWO)	<ul style="list-style-type: none"> <li>Σχεδιασμός ακολουθίας DNA(G. Yang, Wang, Zheng, Zhou, &amp; Zhang, 2017)</li> </ul>

Πίνακας 2 Είδη και εφαρμογές Βιο-εμπνευσμένων Αλγορίθμων (2016-2017)

### Γενετικοί Αλγόριθμοι

Πηγή έμπνευσης για τη δημιουργία των Γενετικών Αλγορίθμων (Genetic Algorithms (GA)) είναι η Δαρβινική θεωρία της εξέλιξης (*evolution*) της φύσης. Σύμφωνα με τους (Mitchell, 1998)(Anci & Dogantekin, 2016)(I.Vlahavas, 2006), οι Γενετικοί Αλγόριθμοι είναι μεθευρετικοί, δηλαδή, έχουν την

ιδιότητα να χρησιμοποιούν την τοπική αναζήτηση (*local search*) βέλτιστης λύσης, ώστε να ξεφύγουν από το τοπικό βέλτιστο (*local optimum*) και να προσδιορίσουν το ολικό βέλτιστο (*global optimum*). Επιπλέον, εκτελούν μία αναζήτηση στο χώρο των υποψήφιων λύσεων, με στόχο την εύρεση αποδεκτών, σύμφωνα με κάποιο κριτήριο, λύσεων. Για να πραγματοποιηθεί ένας γενετικός αλγόριθμος θα πρέπει να υπάρχει ένας μηχανισμός για τη δημιουργία ενός αρχικού πληθυσμού πιθανών λύσεων, μία αναπαράσταση των υποψήφιων λύσεων (συνήθως σε διανυσματική μορφή), μία συνάρτηση καταλληλότητας για την αξιολόγησή τους, ένας μηχανισμός επιλογής γονέων και ένα σύνολο γενετικών τελεστών για τη διαδικασία της αναπαραγωγής. Χρησιμοποιούν πιθανοθεωρητικούς κανόνες μετάβασης και όχι ντετερμινιστικούς. Η εφαρμογή του υλοποιείται με τα παρακάτω βήματα:

- Κωδικοποίηση ενός προβλήματος σε πρόβλημα βελτιστοποίησης
- Δημιουργία ενός συνόλου τυχαίων λύσεων (πληθυσμός)  $P$ , με  $N$  πιθανές λύσεις. (Οι λύσεις αυτές κατά βάση δεν είναι αποδεκτές)
- Κάθε λύση αναπαρίσταται από ένα (*individual*)
- Κάθε άτομο αναπαρίσταται από ένα χρωμόσωμα (*chromosome*) το οποίο είναι μία αλληλουχία συμβόλων 0 ή 1. Κάθε στοιχείο του χρωμοσώματος, που αναπαριστά έναν παράγοντα προς βελτιστοποίηση, ονομάζεται γονίδιο (*gene*). Οι διαφορετικές τιμές που μπορεί να λάβει ένα γονίδιο από ένα πεπερασμένο αλφάβητο ονομάζονται αλληλόμορφες (*alleles*)
- Μετατροπή του πλήθους λύσεων σε διανυσματική μορφή(προτιμάται)
- Χρήση συνάρτησης καταλληλότητας  $f(x)$  (*fitness function*) για αξιολόγηση κάθε μίας ομάδας λύσεων
- Σχηματισμός  $N/2$  ζευγαριών, με τις πιο κατάλληλες λύσεις
- Συνδυασμός ζευγαριού (*mates*) και δημιουργία απογόνων (*offsprings*)
- Υπολογισμός καταλληλότητας απογόνων με τη χρήση συνάρτησης  $f(x)$
- Δημιουργία νέου πληθυσμού, ο οποίος είναι μία βελτιωμένη έκδοση του προηγούμενου
- Τερματισμός διαδικασίας με την εύρεση της πιο κατάλληλης λύσης, ή με τη σύγκλιση όλων των λύσεων σε μία.

### Εξελικτικές Στρατηγικές

Η πρώτη έκδοση του μεθευρετικού αλγορίθμου Εξελικτικών Στρατηγικών (Evolutionary Strategies (ES)), σύμφωνα με τους (Beyer & Schwefel, 2002), ήταν η διμελής (*two-membered ES*), καθώς συμμετείχαν δύο μόνο άτομα, ένας γονέας και ένας απόγονος σε κάθε μία γενιά. Ο απόγονος συνήθως δημιουργείται με τη βοήθεια διωνυμικών κατανομών στο γονέα, με αναμενόμενη τιμή μηδέν και διασπορά  $\sigma^2$ . Η διωνυμική κατανομή είναι μία διακριτή συνάρτηση κατανομής τυχαίας μεταβλητής. Εφαρμόζεται σε τυχαία πειράματα που επαναλαμβάνονται συγκεκριμένες φορές και έχουν δύο δυνατά αποτελέσματα, την επιτυχία ή την αποτυχία. Το άτομο που είναι πιο κατάλληλο χρησιμοποιείται ως



γονέας για την επόμενη γενιά. Ο αλγόριθμος αυτός έγινε γνωστός ως  $(1 + 1) - ES$ . Σύντομα, η σχέση αυτή έδωσε τα ινία στην πολυμελή εξελικτική στρατηγική (*multimemberedES*), στην οποία συμμετέχουν περισσότεροι από έναν γονείς με περισσότερους από έναν απογόνους σε κάθε μία γενιά. Έτσι, υπάρχουν δύο γενικές κατηγορίες αλγορίθμου:

- $(\mu + \lambda) - ES$ : δημιουργούνται  $\lambda \geq 1$  απόγονοι και για να διατηρηθεί σταθερό το μέγεθος του πληθυσμού, οι χειρότεροι  $\lambda$  απόγονοι απορρίπτονται από το συνολικό πλήθος ( $\mu + \lambda$ )
- $(\mu, \lambda) - ES$ : η νέα γενιά βασίζεται αποκλειστικά στην επιλογή μόνο των  $\lambda$  απογόνων, ενώ οι γονείς δε λαμβάνονται υπόψη, όσο κι αν η καταλληλότητά τους είναι μεγάλη ή μικρή.

Οι πολυμελείς μορφές εισάγουν την έννοια της αυτοπροσαρμογής (*self-adaptation*) των παραμέτρων.

### Εξελικτικός Προγραμματισμός

Ο Εξελικτικός Προγραμματισμός (Evolutionary Programming (EP)), χρησιμοποιείται πιο σπάνια συγκριτικά με τις παραπάνω μεθόδους και η χρήση του γίνεται κατά βάση σε προβλήματα βελτιστοποίησης πραγματικών αριθμών, με συνεχείς παραμέτρους. Αποτελεί ίσως ένα είδος στοχαστικού αλγορίθμου. Για την υλοποίηση, όπως αναφέρει ο (D. Fogel, 1993) χρειάζεται :

- Επιλογή πληθυσμού  $\mu$  ατόμων ( $\mu > 1$ )
- Μετάλλαξη κάθε γονέα για δημιουργία  $\mu$  απογόνων, με τη βοήθεια κατανομών πιθανοτήτων.
- Αξιολόγηση απογόνων και επιλογή  $\mu$  καλύτερων ατόμων από το γενικό σύνολο γονέων και απογόνων

### Αλγόριθμος Διαφορικής Εξέλιξης

Ο Αλγόριθμος Διαφορικής Εξέλιξης (Differential Evolution (DE)), όπως αναφέρεται (Koutny, 2016; Storn & Price, 1997), είναι βασισμένος στην εξέλιξη ενός πληθυσμού και είναι ένας στοχαστικός αλγόριθμος ολικής βελτιστοποίησης. Η χρήση του επιφέρει πολλά πλεονεκτήματα όπως είναι η ταχύτητα, η ευκολία, η άμεση σύγκλιση των τιμών και ο περιορισμένος αριθμός παραμέτρων που χρειάζεται για τον έλεγχο των διεργασιών. Υπάρχουν πολλές παραλλαγές ως προς τη χρήση του αλγορίθμου αυτού και η επιλογή της κατάλληλης εκδοχής εξαρτάται από την ίδια τη φύση του προβλήματος. Τα βήματα της εφαρμογής του αλγορίθμου είναι τα εξής:

- Δημιουργία ενός διανυσματικού πληθυσμού σε μία καθορισμένη επιτρεπόμενη παραμετρική περιοχή (αρχικοποίηση)
- Τυχαία επιλογή διανυσμάτων από την περιοχή αυτή και παραγωγή ενός πληθυσμού (μετάλλαξη)
- Διασταύρωση μεταλλαγμένου διανύσματος με διάνυσμα-στόχο
- Αξιολόγηση τιμών συναρτήσεων όλων των δοκιμαστικών διανυσμάτων
- Επιλογή των κατάλληλων διανυσμάτων κατόπιν σύγκρισης κάθε δοκιμαστικού διανύσματος με το αντίστοιχο διάνυσμα-στόχο
- Επανάληψη διαδικασίας για όλα τα μέλη του πληθυσμού

- Τερματισμός με την εύρεση της καλύτερης λύσης ή λόγω κάποιου καθορισμένου κριτηρίου όπως ο αριθμός γενεών, η σύγκλιση/απόκλιση διανυσμάτων και ο υπολογιστικός χρόνος.

•

### **Αλγόριθμοι Γενετικού Προγραμματισμού**

Ο Γενετικός Προγραμματισμός (Genetic Programming Algorithms(GP)) είναι μία ειδική περίπτωση των γενετικών αλγορίθμων, όπου έχει ως στόχο την αυτόματη κατασκευή προγραμμάτων υπολογιστών. Σύμφωνα με τους (I.Vlahavas, 2006; Koza, 1994), οι υποψήφιες λύσεις στον εξελισσόμενο πληθυσμό είναι προγράμματα υπολογιστών και όχι bit-strings. Ο πληθυσμός των υποψήφιων λύσεων απαρτίζεται από τα δέντρα και για να υλοποιηθεί χρησιμοποιεί τις διαδικασίες της αναπαραγωγής, διασταύρωσης και μετάλλαξης. Πιο συγκεκριμένα, παρακάτω παρουσιάζονται τα βήματα εκτέλεσης:

- Δημιουργία τυχαίου αρχικού πληθυσμού και έλεγχος καταλληλότητας με εφαρμογή του προγράμματος σε ένα μεγάλο σύνολο δοκιμαστικών δεδομένων
- Επιλογή ενός ή δύο προγραμμάτων, βάσει του προηγούμενου βήματος
- Δημιουργία νέων προγραμμάτων μέσω διασταύρωσης, επιλογής, δηλαδή, με τυχαίο τρόπο του υποδένδρου ενός γονέα και αντικατάστασής του με ένα υπόδενδρο του άλλου γονέα
- Μετάλλαξη ενός απογόνου για το νέο πληθυσμό
- Επανάληψη αλγορίθμου μέχρι να ικανοποιηθεί ένα κριτήριο τερματισμού
- Τερματισμός διαδικασίας έπειτα από την εύρεση της καλύτερης λύσης.

### **Αλγόριθμοι Βελτιστοποίησης Αποικιών των Μυρμηγκιών**

Ο μεθευρετικός (*meta-heuristics*) Αλγόριθμος Βελτιστοποίησης Αποικιών των Μυρμηγκιών (Ant Colony Optimization Algorithm) προέρχεται από τις αποικίες των μυρμηγκιών (AbWahab, Nefti-Meziani, & Atyabi, 2015; Brownlee, 2011). Τα ζώα αυτά δημιουργούν αγέλες για να κνηγήσουν, να χτίσουν γέφυρες, να βρουν τροφή και να κατασκευάσουν τη φωλιά τους. Σε πολλά είδη μυρμηγκιών, υπάρχει μία ουσία που τοποθετείται στο έδαφος από τα ίδια τα μυρμηγκία καθώς κατευθύνονται στην τροφή τους, η οποία ονομάζεται φερομόνη. Κάποια μυρμηγκία μπορούν να ανιχνεύσουν αυτή την ουσία και έτσι να κατευθυνθούν μέσω αυτής προς την περιοχή με τη μεγαλύτερη συγκέντρωση φερομόνης. Η φερομόνη που υπάρχει στο έδαφος διαμορφώνει το ίχνος φερομονών (*trail*), το οποίο επιτρέπει στα μυρμηγκία να βρίσκουν τις καλές πηγές τροφίμων που έχουν προσδιοριστεί προηγουμένως από άλλα μυρμηγκία. Ο στόχος των τεχνητών μυρμηγκιών είναι παρόμοιος με αυτόν των πραγματικών μυρμηγκιών, δηλαδή να βρουν το συντομότερο δρόμο μεταξύ των κόμβων του γράφου στον οποίο απεικονίζεται το πρόβλημα που εξετάζεται. Τα γενικά βήματα που πρέπει κάποιος να ακολουθήσει για τον ACO αλγόριθμο παρουσιάζονται παρακάτω.

- Απεικόνιση χώρου λύσεων με κατάλληλο γράφο του προβλήματος
- Αρχικοποίηση των παραμέτρων του αλγορίθμου ACO
- Παρακολούθηση πορείας μυρμηγκιών πρακτόρων και δημιουργία λύσεων βάσει αυτής

- Επανακαθορισμός τιμής φερομόνης μυρμηγκιών
- Επανάληψη διαδικασίας μέχρι ικανοποίηση κριτηρίων και εύρεση επιθυμητής λύσης
- Τερματισμός αλγόριθμου με την εύρεση της βέλτιστης λύσης

### Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων

Ένα σμήνος είναι ένας μεγάλος αριθμός από ομοιογενείς παράγοντες οι οποίοι αλληλεπιδρούν τόσο μεταξύ τους όσο και με το περιβάλλον στο οποίο ζουν προκειμένου να επιτελέσουν ένα συγκεκριμένο στόχο, όπως την αναζήτηση τροφής και καταφυγίου (Shi & Eberhart, 1999; C. Yang & Simon, 2005). Ο μεθευρετικός Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization Algorithm (PSO)) μιμείται την κοινωνική συμπεριφορά των ζώων αυτών. Λειτουργεί σε αναλογικά συστήματα, σε διακριτά συστήματα αλλά και σε συνδυασμό αυτών των δύο. Τα γενικότερα βήματα παρουσιάζονται παρακάτω.

- Επιλογή αριθμού σμηνών και σωματιδίων ανά σμήνος
- Αρχικοποίηση θέσεων και ταχυτήτων σμήνους, με επιλογή τυχαίας θέσης για κάθε σωματίδιο
- Αξιολόγηση και σύγκριση κάθε μεμονωμένου σωματιδίου
- Επιλογή καλύτερης θέσης
- Υπολογισμός νέων θέσεων και ταχυτήτων
- Εύρεση καλύτερης δυνατής λύσης
- Εύρεση βέλτιστου σωματιδίου ολόκληρου του σμήνους
- Τερματισμός αλγόριθμου

Εάν η τρέχουσα τιμή είναι καλύτερη από την τιμή του δείγματος, τότε αλλάζουν οι τιμές  $x_{id}$  και  $p_{id}$  και αντίστοιχα μεταβάλλεται και η τιμή της μεταβλητής  $v_{ij}$ . Ο υπολογισμός της ταχύτητας και της θέσης οντοτήτων γίνεται κατανοητός με τις παρακάτω εξισώσεις.

$$v_{ij} = w \cdot v_{id} + c_1 r (p_{id} - x_{id}) + c_2 r \cdot (p_{gd} - x_{id}) \quad (3.1)$$

$$x_{id} = x_{id} + v_{id} \quad (3.2)$$

όπου  $v_{ij} = v_{i1}, v_{i2}, \dots, v_{iD}$  αντιπροσωπεύει το ρυθμό της μεταβολής της θέσης, δηλαδή την ταχύτητα (*velocity*),  $x_{id}$  η θέση,  $p_{id}$  η καλύτερη προηγούμενη θέση,  $g$  η καλύτερη οντότητα του πληθυσμού,  $r$  τυχαία μεταβλητή που ορίζεται για τη συνάρτηση  $\text{rand}()$  με σκοπό την παραγωγή ψευδοτυχαίων αριθμών και  $w$  ο συντελεστής αδράνειας (*inertia weight*). Μεγάλη τιμή του βάρους αυτού υποδηλώνει ολική αναζήτηση ενώ μικρή τιμή του βάρους ανάγει την αναζήτηση σε τοπική.

### Αλγόριθμος Τεχνητής Αποικίας Μελισσών

Ο Αλγόριθμος Τεχνητής Αποικίας Μελισσών (Artificial Bee Colony Optimization Algorithm (ABC)) είναι ένας μεθευρετικός αλγόριθμος βελτιστοποίησης και είναι εμπνευσμένος από τη συμπεριφορά ενός σμήνους μελισσών. Οι μέλισσες αποχωρούν από μία κυψέλη για την αναζήτηση τροφής και όταν βρουν την τροφή, επειδή δεν μπορούν να την συλλέξουν και να την μεταφέρουν μόνες τους, επιστρέφουν στην κυψέλη, ξεφορτώνουν το νέκταρ που πήραν μαζί τους και ενημερώνουν τις υπόλοιπες μέλισσες για την πηγή

που ανακάλυψαν. Η ενημέρωση γίνεται με ειδικές κινήσεις που λέγονται waggled dance. Όπως αναφέρουν οι (Dervis Karaboga et al., 2014) υπάρχουν κυρίως τρεις ομάδες μελισσών:

- Οι εξερευνητριες μέλισσες (*employed bees*) που ασχολούνται με συγκεκριμένες πηγές τροφής που επιλέγουν από ένα καθορισμένο σύνολο, και στη συνέχεια μοιράζονται την πληροφορία αυτή μέσω της ειδικής κίνησής τους με τις μέλισσες της κυψέλης. Οι πιθανές πηγές τροφής αντιπροσωπεύουν τις πιθανές λύσεις στο πρόβλημα
- Οι θεατές μέλισσες (*onlooker bees*) που περιμένουν πίσω στην κυψέλη, ενημερώνονται από τις εξερευνητριες μέλισσες και αναζητούν μία καλύτερη γειτονική πηγή τροφής
- Οι ανιχνεύτριες μέλισσες (*scout bees*) είναι οι εξερευνητριες που αναζητούν τυχαία νέες πηγές τροφίμων

Η θέση της πηγής τροφής στον αλγόριθμο αντιπροσωπεύει μία πιθανή λύση ενώ η ποσότητα του νέκταρ αντιστοιχεί στην ποιότητα (fitness) της λύσης.

Παρακάτω παρουσιάζεται συνοπτικά η διαδικασία εφαρμογής του αλγορίθμου:

- Τυχαία επιλογή ενός συνόλου πηγών τροφής-πιθανών λύσεων (το πλήθος πηγών τροφής πρέπει να υπερτερεί αριθμητικά από το πλήθος των μελισσών εξερευνητριών)
- Υπολογισμός τιμής αντικειμενικής συνάρτησης και αντιστοίχιση κάθε πηγής τροφής σε μία εξερευνητριά μέλισσα
- Επιλογή πηγής τροφής από θεατές μέλισσες βασισμένες από πληροφορία που έχουν λάβει (αναπαραγωγή), με πιθανότητα

$$P_i = \frac{fit}{\sum_{n=1}^N fit_n} \quad (3.3)$$

όπου  $N$  ο αριθμός των πηγών τροφής,  $i$  η θέση, η οποία είναι ανάλογη με το πόσο νέκταρ υπάρχει σε κάθε πηγή τροφής και  $fit$  η τιμή της αντικειμενικής συνάρτησης καταλληλότητας.

- Αντιστοίχιση κάθε πηγής τροφής με μία συγκεκριμένη μέλισσα. Αν για κάποια πηγή τροφής βρεθεί από τις θεατές μέλισσες μία καλύτερη πηγή τροφής, τότε η πηγή τροφής αντικαθίσταται στην μνήμη των μελισσών από την καλύτερή δυνατή πηγή τροφής.

Θα πρέπει να επισημανθεί πως οι μέλισσες δεν είναι λύσεις αλλά διαδικασίες που εφαρμόζονται σε κάποια λύση (πηγή τροφής), με στόχο να την βελτιώσουν. Έτσι, το πλήθος των μελισσών που ακολουθούν τη μέλισσα εξερευνητριά που βρήκε μία πηγή τροφής, αντιστοιχεί στο πλήθος δοκιμών αναζήτησης για τη νέα πηγή.

### **Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων**

Όταν μία πυγολαμπίδα θέλει να προσελκύσει κάποια άλλη ή να προσεγγίσει κάποιο θήραμα εκκρίνει λουσιφερίνη (Krishnanand & Ghose, 2009). Όσο πιο έντονη είναι η λάμψη, τόσο πιο ελκυστική γίνεται απέναντι στις υπόλοιπες. Έτσι, ο βαθμός της ελκυστικότητας μίας πυγολαμπίδας είναι ανάλογος της φωτεινότητάς της. Ο μεθευρετικός Αλγόριθμος Βελτιστοποίησης Σμήνους Πυγολαμπίδων (Glowworm Swarm based Optimization algorithm (GSO)) περιγράφεται από τα παρακάτω βήματα:

- Τυχαία τοποθέτηση η οντοτήτων στο χώρο λύσεων
- Πραγματοποίηση κίνησης από κάθε οντότητα βάσει της έντασης του σήματος που λαμβάνει από κάποιο γείτονά της
- Υπολογισμός αντικειμενικής συνάρτησης  $f(x(t))$  και τιμή λάμψης πυγολαμπίδας
- Επιλογή γειτόνων, με τη χρήση πιθανοτήτων, προς τους οποίους θα κινηθεί η πυγολαμπίδα
- Ενημέρωση της λάμψης και της κίνησης, έπειτα από κάθε επανάληψη του αλγορίθμου. (Η κατεύθυνση καθορίζεται βάσει της τιμής της λάμψης της πυγολαμπίδας. Η κίνηση πραγματοποιείται προς τις υψηλότερες σε τιμή λάμψεις πυγολαμπίδες).

### **Μέθοδος Αναζήτησης Τροφής Βακτηρίων**

Ο μεθυστικός Αλγόριθμος Μεθόδου Αναζήτησης Τροφής Βακτηρίων (Bacterial Foraging Optimization Algorithm (BFOA)) συνδυάζει την μικροβιολογία και την μηχανική (Binitha, Sathya, & others, 2012b). Προέρχεται από την μοντελοποίηση της κίνησης του βακτηρίου E.coli προκειμένου να βρει τροφή. Αποτελείται κυρίως από τρεις μηχανισμούς-φάσεις:

- Χημειοταξία (*chemotaxis*)
- Αναπαραγωγή (*reproduction*)
- Εξάλειψη- διασπορά (*elimination-dispersal*)

Η πρώτη φάση είναι η κυτταρική κίνηση και περιλαμβάνει τη δραστηριότητα των βακτηρίων που συγκεντρώνονται σε περιοχές πλούσιες σε θρεπτικά υλικά, μέσω ενός ενδοκυτταρικού (*cell-to-cell*) μηχανισμού επικοινωνίας. Στη δεύτερη φάση, την αναπαραγωγή, έχουν απομείνει μόνο τα βακτήρια που έχουν προσαρμοστεί καλύτερα στις συνθήκες και μόνο αυτά μεταβιβάζουν τα γενετικά χαρακτηριστικά τους. Τα υπόλοιπα βακτήρια, δηλαδή αυτά που είναι λιγότερο προσαρμοσμένα στις συνθήκες, αρχίζουν να εξαφανίζονται. Στην τελευταία φάση, ο αλγόριθμος εξασφαλίζει ποικιλομορφία στα είδη και βελτίωση στην ικανότητα της αναζήτησης (*global search*), επιλέγοντας τυχαία τμήματα του πληθυσμού των βακτηρίων.

Η διαδικασία του αλγορίθμου περιγράφεται από τα εξής βήματα:

- Αρχικοποίηση του πληθυσμού, έχοντας τυχαία κατανομή στο χώρο
- Υπολογισμός απόδοσης, σύμφωνα με τη φύση του προβλήματος, για τον πληθυσμό
- Επανάληψη αλγορίθμου μέχρι να προκύψει καλύτερη λύση
- Τερματισμός αλγορίθμου

### **Αλγόριθμος Νυχτερίδας**

Ο Αλγόριθμος Νυχτερίδας (Bat Algorithm (BA)) είναι μεθυστικός και εμπνευσμένος από τον τρόπο που οι νυχτερίδες *microbat* εντοπίζουν την τροφή τους αλλά και αντιλαμβάνονται τον χώρο (X.-S. Yang, 2010). Η ιδιότητα του ηχοεντοπισμού, που χρησιμοποιούν οι νυχτερίδες κατά την αναζήτηση της τροφής τους, είναι αυτή που προσομοιώνεται για την δημιουργία του αλγορίθμου. Για την υλοποίηση του αλγορίθμου τα βήματα είναι:

- Εύρεση αντικειμενικής συνάρτησης
- Αρχικοποίηση πληθυσμών και απαιτούμενων τιμών
- Παραγωγή νέων λύσεων από νέες τροχιές και θέσεις
- Επιλογή καλύτερης λύσης και παραγωγή λύσεων βάσει αυτής
- Παραγωγή νέας λύσης
- Αξιολόγηση νυχτερίδων και εντοπισμός καλύτερης
- Τερματισμός αλγορίθμου

### Αλγόριθμος Επιλογής Κλώνων

Ο μεθευρετικός Αλγόριθμος Επιλογής Κλώνων (Clonal Selection Algorithm (CSA)) είναι ένας από τους βασικότερους αλγόριθμους της θεωρίας Τεχνητών Ανοσοποιητικών Συστημάτων (Artificial Immune Systems (AIS)) και χρησιμοποιείται ευρέως για προβλήματα βελτιστοποίησης (De Castro & Von Zuben, 2000).

Το ανοσοποιητικό σύστημα είναι ένα όργανο του ανθρώπινου σώματος που έχει ως ρόλο να προστατεύει τον οργανισμό από βλαβερούς εξωτερικούς παράγοντες που εισέρχονται ανεπιθύμητα, όπως τα βακτήρια και ιοί. Σύμφωνα με τον (Talbi, 2009) για να πραγματοποιηθεί η επίλυση ενός προβλήματος βελτιστοποίησης με κάποιον αλγόριθμο εμπνευσμένο από το ανοσοποιητικό σύστημα, θα πρέπει να αντιστοιχηθούν οι απαιτούμενοι όροι. Έτσι, παρακάτω παρουσιάζεται η ορολογία αυτή:

- Το αντίσωμα, αντιστοιχεί στη λύση του προβλήματος
- Η συγγένεια (*affinity*) αντιστοιχεί στην αντικειμενική συνάρτηση για την καταλληλότητα
- Το αντιγόνο αντιστοιχεί στο πρόβλημα προς επίλυση
- Η κλωνοποίηση αντιστοιχεί στην παραγωγή λύσεων
- Η σωματική μετάλλαξη ή υπερμετάλλαξη (*Somaticmutation (hypermutation)*) αντιστοιχεί στη διαδικασία της πολλαπλής μετάλλαξης μίας λύσης
- Η ωρίμανση της συγγένειας (*Affinitymaturation*) αντιστοιχεί στην επιλογή βέλτιστων λύσεων για το πρόβλημα
- Η διαδικασία διόρθωσης των υποδοχέων (*Receptor editing*) αντιστοιχεί στη διασπορά των λύσεων σε όλο το χώρο λύσεων του προβλήματος (*Diversification*)

Ο αλγόριθμος υλοποιείται ακολουθώντας τα παρακάτω βήματα:

- Αρχικοποίηση του πληθυσμού αντισωμάτων σύμφωνα με τη φύση του προβλήματος
- Τυχαία επιλογή αντιγόνου, υπολογισμός αντικειμενικής συνάρτησης
- Επιλογή καλύτερων αντισωμάτων από τον πληθυσμό
- Κλωνοποίηση και μετάλλαξη λύσεων για δημιουργία του νέου πληθυσμού
- Επιλογή των καλύτερων κλώνων και αντικατάσταση αρχικού πληθυσμού με αυτούς
- Δημιουργία τυχαίων αντισωμάτων και αντικατάσταση παλαιότερων

### **Αλγόριθμος Αρκτικής Γαρίδας**

Η δημιουργία του ευριστικού Αλγορίθμου Αρκτικής Γαρίδας (KrillHerd (KH)) βασίζεται στο γεγονός ότι η αρκτική γαρίδα σχηματίζει σμήνος/κοπάδι για άμυνα αλλά και για εύρεση τροφής (A. Gandomi & Alani, 2012). Οι αρχικές συνθήκες καθορίζονται από την κατάσταση που το κοπάδι δέχεται. Για την καλύτερη απόδοση η θέση του αναζητητή προσδιορίζεται από τους παρακάτω παράγοντες:

- Κίνηση των υπόλοιπων αναζητητών
- Δραστηριότητα τροφοληψίας (*foraging*)
- Τυχαία διάχυση

Τα βήματα που περιγράφουν τον αλγόριθμο είναι τα παρακάτω:

- Αρχικοποίηση μεταβλητών προβλήματος
- Δημιουργία του πληθυσμού
- Αξιολόγηση απόδοσης κάθε αναζητητή σύμφωνα με την θέση του
- Υπολογισμός κίνησης και διάχυσης
- Καθορισμός νέων θέσεων αναζητητών
- Επανάληψη μέχρι εκπλήρωσης κριτηρίου τερματισμού
- Τερματισμός Αλγορίθμου

### **Αλγόριθμος Βελτιστοποίησης βασισμένος στη Βιογεωγραφία**

Ο Αλγόριθμος Ολικής Βελτιστοποίησης βασισμένος στη Βιογεωγραφία (Biogeography -based Optimization Algorithm (BBO)) δημιουργήθηκε με πηγή έμπνευσης τη θεωρία της βιογεωγραφίας, που παρουσιάστηκε πρώτη φορά από τον Dan Simon (Simon, 2008). Η βιογεωγραφία είναι η μελέτη της γεωγραφικής κατανομής των διαφόρων οργανισμών στη φύση με την πάροδο του χρόνου, στους διάφορους οικοτόπους. Δύο βασικές έννοιες που κατέχουν κεντρικό ρόλο στον αλγόριθμο αυτό είναι η μετανάστευση και η μετάλλαξη. Οι πιθανές λύσεις ενός προβλήματος αναπαρίστανται με διανυσματική ακέραια μορφή. Ο κάθε ένας ακέραιος αριθμός στο διάνυσμα θεωρείται μία μεταβλητή δείκτη καταλληλότητας (*Suitability Index Variable (SIV)*). Οι πιο κατάλληλες λύσεις θεωρούνται βιότοποι με υψηλό δείκτη καταλληλότητας (*Highhabitat Suitability Index (HSI)*), ενώ χαμηλό δείκτη καταλληλότητας (*Low habitat Suitability Index (LSI)*) έχουν οι λύσεις που δεν είναι οι πιο αντιπροσωπευτικές. Μία γενικότερη περιγραφή του αλγορίθμου είναι η παρακάτω:

- Αρχικοποίηση των παραμέτρων
- Αρχικοποίηση ενός τυχαίου συνόλου βιοτόπων, με κάθε βιότοπο να αντιστοιχεί σε μία πιθανή λύση του προβλήματος
- Καθορισμός μεταβλητών για κάθε ένα βιότοπο, υπολογισμός του δείκτη HSI
- Μετάλλαξη συγκεκριμένων βιοτόπων και υπολογισμός νέων δεικτών HSI
- Επανάληψη αλγορίθμου για καθορισμένο αριθμό γενεών
- Τερματισμός έπειτα από εύρεση κατάλληλης λύσης ή ικανοποίηση κάποιας συνθήκης.

### Αλγόριθμος Βελτιστοποίησης Διηθητικών Ζιζανίων

Ο μεθευρετικός Αλγόριθμος Βελτιστοποίησης Διηθητικών Ζιζανίων (Invasive Weed Optimization Algorithm (IWO)) αντλεί την έμπνευσή του από τον τρόπο με τον οποίο τα ανεπιθύμητα φυτά, τα οποία μπορεί να είναι ζιζάνια ή αγριόχορτα, δημιουργούν αποικίες και αναπαράγονται (Mehrabian & Lucas, 2006). Ο όρος ζιζάνιο (*weed*) αντιστοιχεί σε μία λύση του προβλήματος, ενώ ο όρος σπόρος (*seed*) προκύπτει με κάποια στοχαστική μέθοδο από τον *weed* και εκφράζει μία πιθανή λύση του προβλήματος. Τα γενικά βήματα υλοποίησης του αλγορίθμου IWO είναι τα παρακάτω:

- Αρχικοποίηση του πληθυσμού *seed* που προέρχονται από μία τυχαία κατανομή στο χώρο αναζήτησης
- Κάθε ένα *seed*, που εντοπίζεται εντός του χώρου αναζήτησης, αντιστοιχίζεται με μία τιμή καταλληλότητας (*fitness*) και μετατρέπεται σε αναπαραγωγή των ζιζανίων (*reproduction*)
- Τα νέα *weed* που αναπτύσσονται, διασκορπίζονται τυχαία στο χώρο αναζήτησης (*spatial dispersal*)
- Επανάληψη του αλγορίθμου μέχρι την καθορισμένη τιμή
- Τερματισμός αλγορίθμου, όταν το *weed* βρεθεί με την καλύτερη τιμή καταλληλότητας



# 4.Αλγόριθμος Βελτιστοποίησης Καταιγισμού

## Ιδεών

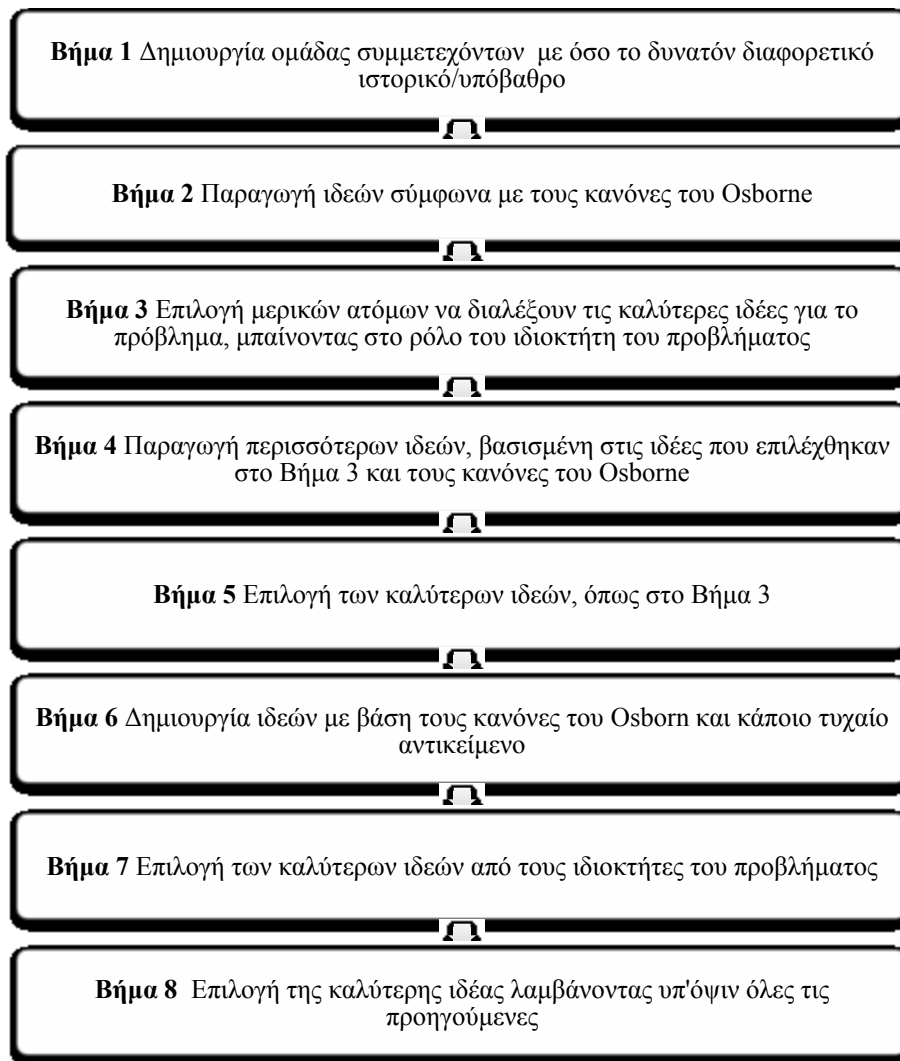
### 4.1 Η διαδικασία καταιγισμού ιδεών

Ο εγκέφαλος είναι υπεύθυνος για πολλές από τις λειτουργίες του σώματος. Μία από τις πολυάριθμες δράσεις του εγκεφάλου είναι η παραγωγή ιδεών. Ο ομαδοσυνεργατικός ιδεοκαταιγισμός (brainstorming process), μπορεί να επιφέρει πολλά πλεονεκτήματα τόσο στη ρεαλιστική ζωή όσο και αλγοριθμικά. Πιο συγκεκριμένα, ένα πρόβλημα κάποιες φορές είναι δύσκολο να λυθεί από το ίδιο το άτομο που το έχει, όμως μία ομάδα ατόμων με διαφορετικές απόψεις και προσωπικότητες είναι πιθανότατα πιο αποτελεσματικό να επιφέρει καλύτερες ιδέες και λύσεις. Αυτή η διεργασία αποτελεί την πηγή έμπνευσης για τη δημιουργία του αντίστοιχου Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών, που μελετάται στην παρούσα ενότητα.

Η διαδικασία του καταιγισμού ιδεών για να πραγματοποιηθεί, χρειάζεται:

- Συντονιστή (*facilitator*): κατευθύνει τη διαδικασία ώστε να προχωράει εποικοδομητικά. Ο ίδιος έχει ως καθήκον να συγκεντρώνει τις νέες παραγόμενες και ανοργάνωτες ιδέες σε έναν πίνακα ή ένα χαρτί, ώστε να εξετάζονται στη συνέχεια με μεγαλύτερη ευκολία, όμως, θα πρέπει να αποφευχθεί η ενεργή συμμετοχή του στη συζήτηση.
- Μέλη/συμμετέχοντες (*group of brainstorming people*): συζητούν και παράγουν ιδέες σύμφωνα με την καθοδήγηση του συντονιστή κατά τη διάρκεια της διαδικασίας καταιγισμού ιδεών
- Άτομα με πρόβλημα (*problem owners*): είναι υπεύθυνα να επιλέξουν τις ιδέες που θεωρούν καλύτερες σε κάθε επανάληψη της διαδικασίας του ιδεοκαταιγισμού

Κατά τη διαδικασία του καταιγισμού ιδεών όλα τα μέλη μαζεύονται, συζητούν, γενικεύουν τις ιδέες τους ακολουθώντας τους κανόνες του Osborn, που αναφέρονται στη συνέχεια, ανταλλάζουν απόψεις και εν τέλει η διαδικασία τελειώνει έχοντας βρει την πιο κατάλληλη λύση (Shi, 2011).



Πίνακας 3 Τα βήματα διεξαγωγής της διαδικασίας καταιγισμού ιδεών

## 4.2 Κανόνες του Osborn

Ο καταιγισμός ιδεών ως τεχνική δημιουργήθηκε από το διαφημιστικό στέλεχος, συγγραφέα και θεωρητικό δημιουργικότητας (*creativity theorist*) Alex F. Osborn. (Shi, 2011) Μπορεί να εφαρμοστεί σε οποιοδήποτε τομέα της ζωής, με ιδιαίτερο πλεονεκτικό ρόλο στη διδασκαλία. Βασίζεται στη συμμετοχή, στην ελεύθερη και αυθόρμητη έκφραση ιδεών και στη διαφορετικότητα των ανθρώπων. Σύμφωνα με τον Osborn οι κανόνες για σωστό καταιγισμό ιδεών είναι οι εξής:

- Αναστολή κριτικής (*Suspend Judgment*)  
Δεν υπάρχει καλή ή κακή ιδέα, καθώς όλες οι ιδέες είναι ενδιαφέρουσες. Είναι άωφελο λοιπόν, να υπάρχει σχολιασμός κατά τη διάρκεια που ένα άτομο εκφράζεται, γι ' αυτό και πρέπει να αποφεύγεται η κριτική κατά τη διάρκεια της διαδικασίας του καταιγισμού ιδεών
- Ίση αξία ιδεών (*Anything goes*)  
Οτιδήποτε έρθει στο μυαλό ενός ατόμου κατά τη διάρκεια της συνδιάσκεψης με τους υπολοίπους, θα ήταν πλεονεκτικό να το μοιραστεί και όχι να αγνοηθεί. Οι ασυνήθιστες και διαφορετικές απόψεις πρέπει να αποδέχονται και να επεξεργάζονται, πριν προδικαστούν.

- Ποσότητα ιδεών (*Go for quantity*)  
Για να επιτευχθεί η ποιότητα των ιδεών, ο Osborn ισχυρίζεται πως θα πρέπει να υπάρχει ένα μεγάλο πλήθος ιδεών, μέσα από τις οποίες θα διακριθούν οι καλύτερες και πιο κατάλληλες για το πρόβλημα.
- Συνδυασμός και βελτίωση ιδεών (*Cross-fertilize (Piggyback)*)  
Πολλές φορές μία ιδέα μπορεί να αποτελέσει ένα γερό θεμέλιο ώστε αν αναβαθμιστεί και να προκύψουν πολλές καινοτόμες σκέψεις από αυτήν.

### 4.3 Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών

Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών (Brain Storm Optimization Algorithm (BSO)) ανήκει στην κατηγορία Αλγορίθμων Σμήνου(Πίνακας 2) και είναι εμπνευσμένος, όπως υποδηλώνει και το όνομά του, από τη διαδικασία που ακολουθεί ο άνθρωπος, κατά τη διάρκεια παραγωγής ιδεών. Υπάρχουν δύο χαρακτηριστικά που κάνουν τον αλγόριθμο να ξεχωρίζει:

- **Συγκλίνουσα λειτουργία (*Convergent Operation*):** Σχεδιασμένη να μιμείται την κατεργασία όπου οι ιδιοκτήτες των προβλημάτων διαλέγουν τις καλύτερες δυνατές ιδέες, έτσι ώστε στην επόμενη γενιά να επικεντρωθεί η παραγωγή ιδεών στις πιο πρόσφορες υπάρχουσες λύσεις. Γενικότερα, η διαδικασία αυτή έχει ως αποτέλεσμα από ένα μεγάλο πληθυσμό να διαχωριστεί ένα συγκεκριμένο σύνολο πιθανών λύσεων.  
Για να επιτευχθεί αυτή η λειτουργία στον αυθεντικό αλγόριθμο χρησιμοποιείται η ομαδοποίηση K-μέσων κατά την οποία διάφορα άτομα ομαδοποιούνται σε διαφορετικές συστάδες, με την καλύτερη ιδέα/ λύση σε κάθε μία συστάδα να χρησιμοποιείται για να εκπροσωπήσει την αντίστοιχη καλύτερη ιδέα που διαλέγει ο ιδιοκτήτης του προβλήματος προκειμένου να επιλυθεί το ζήτημα που τον απασχολεί.
- **Αποκλίνουσα λειτουργία (*Divergent Operation*):** Κατά τη διαδικασία αυτή παράγονται ιδέες βασισμένες σε προηγούμενες ιδέες .  
Στην πραγματικότητα, όσον αφορά την υλοποίηση, προστίθενται γκαουσιανές τυχαίες τιμές σε μία προϋπάρχουσα ιδέα.(Shi, 2011)(Shi, 2015)

Επιπρόσθετα ο αλγόριθμος BSO χαρακτηρίζεται από τρεις στρατηγικές: την ομαδοποίηση των λύσεων (*solution clustering*), την παραγωγή νέων ιδεών (*new individual generation*) και την επιλογή ιδεών (*selection*).

#### 4.3.1 Ομαδοποίηση λύσεων

Μια από τις βασικές στρατηγικές που χρησιμοποιούνται στον αλγόριθμο καταιγισμού ιδεών είναι η ομαδοποίηση των λύσεων και επιτυγχάνεται με τη χρήση του K-μέσου (Cheng et al., 2017). Με τη

μέθοδο αυτή μπορεί να προσδιοριστεί μία συγκεκριμένη περιοχή αναζήτησης για εύρεση των πιθανών λύσεων. Ειδικότερα, στον αλγόριθμο χρησιμοποιείται η πιθανότητα  $P_{clustering}$  η οποία ελέγχει την πιθανότητα της αντικατάστασης ενός κέντρου μίας συστάδας με μία τυχαία νέα παραγόμενη λύση. Αυτό συνεισφέρει στο να αποφευχθεί η πρόωμη σύγκλιση του αλγορίθμου και συμβάλει θετικά ώστε τα άτομα να "προσπερνούν" ("jump out") πιθανά τοπικά μέγιστα.

### 4.3.2 Παραγωγή νέων ιδεών

Μία ακόμα σημαντική διαδικασία που περιλαμβάνεται στον αλγόριθμο BSO, όπως έχει ήδη αναφερθεί, είναι η παραγωγή νέων πιθανών λύσεων. Όπως φαίνεται και από τον **Πίνακα 4**, η διεργασία αυτή μπορεί να γίνει βασισμένη σε μία ιδέα ή και παραπάνω. Η παραγωγή νέων ιδεών γίνεται βασισμένη στην εξής σχέση:

$$x_{new}^d = x_{sel}^d + \xi \cdot N(\mu, \sigma) \quad (4.1)$$

$$\xi = \log sig \left( \frac{0.5 \cdot \max\_iteration - current\_iteration}{k} \right) \cdot r \quad (4.2)$$

όπου  $x_{new}^d$  d-οστή διάσταση της νέας πιθανής λύσης που δημιουργήθηκε,  $x_{sel}^d$  η d-οστή διάσταση μίας πιθανής λύσης που επιλέχθηκε προκειμένου να παραχθεί μία νέα πιθανή λύση,  $N(\mu, \sigma)$  Γκαουσιανή τυχαία συνάρτηση με μέση τιμή  $\mu$  και διακύμανση  $\sigma$ ,  $\xi$  συντελεστής (*step-size*), σταθμίζει τη συμβολή της τυχαίας γκαουσιανής τιμής,  $r$  τυχαία μεταβλητή που ορίζεται για τη συνάρτηση  $rand()$  με σκοπό την παραγωγή ψευδοτυχαίων αριθμών,  $\log sig$  λογαριθμική σιγμοειδής συνάρτηση μεταφοράς,  $\max\_iteration$  μέγιστος αριθμός επαναλήψεων μίας εκτέλεσης ενός BSO προγράμματος,  $current\_iteration$  αριθμός συγκεκριμένης επανάληψης,  $k$  συντελεστής αλλαγής κλίσης συνάρτησης  $\log sig$ , υπεύθυνος για σύγκλιση αλγορίθμου (*convergence speed*) και  $rand()$  συνάρτηση τυχαίων τιμών με στόχο τη γενίκευση των λύσεων στο  $[0,1]$

Ειδικότερα στον αλγόριθμο χρησιμοποιείται η πιθανότητα  $P_{generation}$  για να καθοριστεί εάν μία νέα λύση θα προκύψει από μία ή δύο προϋπάρχουσες συστάδες (Cheng et al., 2017) (Shi, 2011). Επιπλέον, οι πιθανότητες  $P_{one\_cluster}$ ,  $P_{two\_clusters}$  είναι υπεύθυνες για τον καθορισμό των συστάδων ή των ατόμων που θα επιλεγθούν για την παραγωγή νέων λύσεων βάσει μίας ή δύο συστάδων αντίστοιχα.

### 4.3.3 Επιλογή ιδεών

Η επιλογή συγκεκριμένων ιδεών χρησιμοποιείται για να διατηρούνται οι καλύτερες λύσεις ανά γενιά (Shi, 2011). Ειδικότερα, όσο πραγματοποιούνται οι διεργασίες της ομαδοποίησης και της παραγωγής νέων ιδεών, προστίθενται νέες ιδέες ενώ συλλέγονται οι καλύτερες από τις δυνατές υπάρχουσες. Με αυτόν τον τρόπο εξασφαλίζεται ποικιλομορφία για ολόκληρο τον πληθυσμό. Παρακάτω, στον **Πίνακα 4** παρουσιάζεται η διαδικασία υλοποίησης του αλγορίθμου BSO, που βασίζεται στη διαδικασία καταγισμού ιδεών (**Πίνακας 3**).

Πιο συγκεκριμένα, στον **Πίνακα 5** το **Βήμα 1** αντιστοιχεί στην αρχικοποίηση του αλγορίθμου BSO, το **Βήμα 2**, **Βήμα 3**, **Βήμα 4** εξυπηρετούν το σκοπό των **Βημάτων 3,5,7** αντίστοιχα του **Πίνακα 3**, με σκοπό τη δημιουργία καλύτερων ιδεών.(Shi, 2011) Επιπρόσθετα, το **Βήμα 5** προσομοιώνει το **Βήμα 6** στον **Πίνακα 3**, που αποσκοπεί στη δημιουργία πληθυσμού. Το **Βήμα 6b** έχει ως στόχο τη δημιουργία νέας ιδέας εμπνευσμένης από μία ήδη υπάρχουσα ενώ το **Βήμα6c** προσομοιώνει τη δημιουργία νέας ιδέας από δύο υπάρχουσες ιδέες, προερχόμενες από διαφορετικές ομάδες. Επιπλέον, ο αριθμός των ομάδων αντιπροσωπεύει τα άτομα που έχουν το πρόβλημα (*problem owner*), τα κέντρα των συστάδων προσομοιώνουν τις καλύτερες ιδέες που έχουν επιλεγεί από τα προαναφερθέντα άτομα και τέλος το μέγεθος του πληθυσμού  $n$  προσομοιώνει το πλήθος των ιδεών που δημιουργούνται σε κάθε γενιά. Η μεταβλητή  $n$  προτιμάται να είναι σταθερή και προκαθορισμένη.

Για τις εφαρμογές της παρούσας πτυχιακής εργασίας, οι αρχικοποιήσεις των μεταβλητών γίνονται βάσει της δημοσίευσης του (Shi, 2011), όπως αναγράφονται στον παρακάτω πίνακα:

**Πίνακας 4 Αρχικοποίηση μεταβλητών BSO για SVM**

N	$p_{5a}$	$p_{6b}$	$p_{6biii}$	k	max_iteration	n_c	$\xi$	$\mu$	$\sigma$
20	0.2	0.8	0.4	20	10	5	20	0	1

**Βήμα 1** Τυχαία παραγωγή  $n$  πιθανών λύσεων-άτομα

**Βήμα 2** Συσταδοποίηση  $n$  πιθανών λύσεων σε  $m$  ομάδες

**Βήμα 3** Αξιολόγηση των  $n$  πιθανών ατόμων

**Βήμα 4** Κατάταξη των πιθανών ατόμων  $n$  σε και επιλογή καλύτερου ατόμου κάθε ομάδας ως κέντρο της συστάδας

**Βήμα 5** Τυχαία παραγωγή μίας τιμής μεταξύ 0 και 1. Αν η τιμή αυτή είναι μικρότερη από μία προκαθορισμένη πιθανότητα  $P_{5a}$

a) Τυχαία επιλογή κέντρου συστάδας

b) Τυχαία δημιουργία πιθανού ατόμου ώστε να αντικαταστήσει το επιλεγμένο κέντρο της συστάδας

**Βήμα 6** Δημιουργία νέων πιθανών ατόμων

a) Τυχαία δημιουργία μίας τιμής μεταξύ 0 και 1

b) Εάν η τιμή είναι μικρότερη από μία προκαθορισμένη πιθανότητα  $P_{6b}$

i) Τυχαία **επιλογή μίας συστάδας** με πιθανότητα  $P_{6bi}$

ii) Δημιουργία μίας τυχαίας τιμής μεταξύ 0 και 1

iii) Αν η τιμή είναι μικρότερη από μία πιθανότητα  $P_{6biii}$ , τότε πρέπει να γίνει επιλογή κέντρου συστάδας και προσθήκη τυχαίων τιμών ώστε να προκύψει νέο άτομο

iv) Διαφορετικά, τυχαία επιλογή ενός ατόμου από αυτή τη συστάδα, προσθήκη τυχαίων τιμών και παραγωγή νέου ατόμου.

c) Αλλιώς, τυχαία **επιλογή δύο συστάδων** και παραγωγή νέου ατόμου

i) Δημιουργία τυχαίας τιμής

ii) Αν η τιμή αυτή είναι μικρότερη από μία προκαθορισμένη πιθανότητα  $P_{6c}$ , τα δύο κέντρα των συστάδων συνδυάζονται και έπειτα προστίθενται τυχαίες τιμές για τη δημιουργία νέου ατόμου

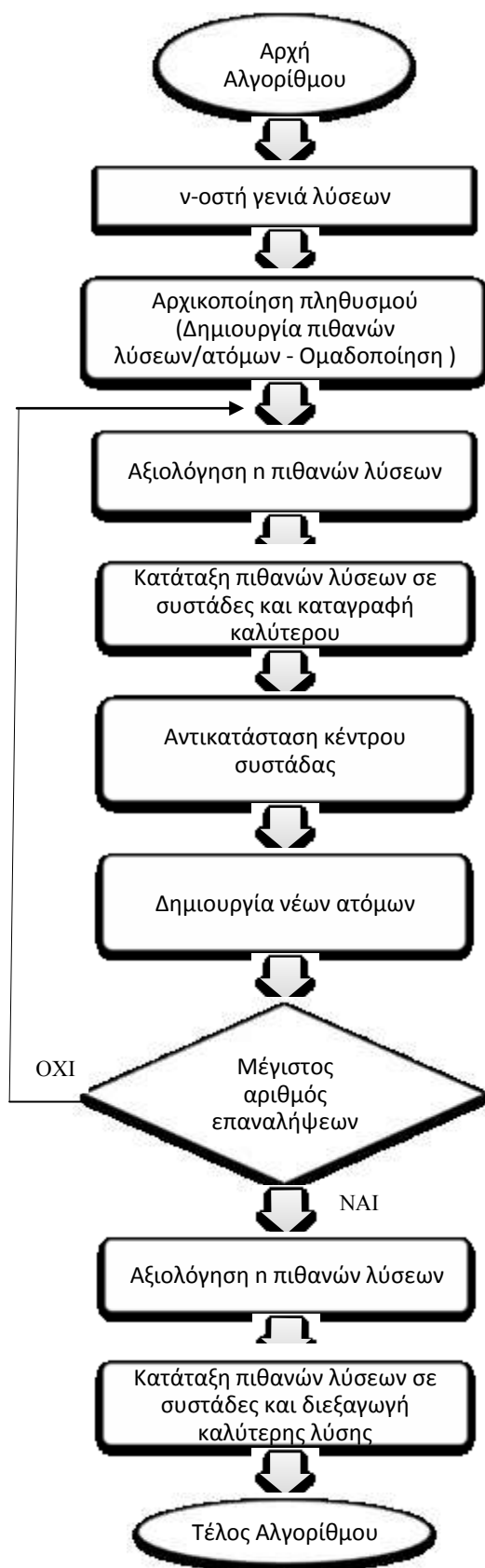
iii) Διαφορετικά, δύο τυχαία επιλεγμένες πιθανές λύσεις από κάθε συστάδα συνδυάζονται και προστίθενται με τυχαίες τιμές ώστε να προκύψει νέο άτομο

d) Το πρόσφατα παραγόμενο άτομο αφού συγκριθεί με τις υπάρχουσες, αν είναι καλύτερη, διατηρείται και καταγράφεται.

**Βήμα 7** Αν έχουν δημιουργηθεί επιτυχώς  $n$  άτομα, μετάβαση στο Βήμα 8. Αν όχι, μετάβαση στο Βήμα 6

**Βήμα 8** Τερματισμός αλγορίθμου αν ο μέγιστος προκαθορισμένος αριθμός επαναλήψεων έχει πραγματοποιηθεί. Αλλιώς, μετάβαση στο Βήμα 2

Πίνακας 5Βήματα υλοποίησης αλγορίθμου BSO



Εικόνα 9 Διάγραμμα Ροής Αλγορίθμου BSO

## 4.4 Βιβλιογραφική ανασκόπηση & Εφαρμογές του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών

Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών βρίσκει πολλές εφαρμογές με ποικίλους τρόπους και σε διάφορους κλάδους επιστημών. Έπειτα από αναζήτηση, κατά βάση, στη Scopus, καταγράφονται στον παρακάτω πίνακα εφαρμογές του αλγορίθμου Brain Storm Optimization κατά τα έτη 2011-2018.

A/A	Εργασία	Χρονολογία	Συνεισφορά/Εφαρμογή	Παραλλαγή BSO
1	(Shi, 2011)	2011	BSO Algorithm	Στην εργασία αυτή προτάθηκε για πρώτη φορά ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών
2	(Zhan, Zhang, Shi, & Liu, 2012)	2012	Modified Brain Storm Optimization Algorithm	<ul style="list-style-type: none"> <li>• Simple Grouping Method (SGM) για την ομαδοποίηση συντελεστών</li> <li>• Idea Difference Strategy (IDS) για τη δημιουργία συντελεστών</li> </ul>
3	(D. Zhou, Shi, & Cheng, 2012)	2012	BSO with modified step-size and individual generation	<ul style="list-style-type: none"> <li>• Προσαρμοστικός συντελεστής παραμέτρου του βήματος</li> <li>• Διαφοροποιημένος αριθμός ατόμων σε κάθε νέα γενιά</li> <li>• Στρατηγική επιλογής ατόμων για την επόμενη παραγωγή νέας γενιάς</li> </ul>
4	(Jadhav, Sharma, Patel, & Roy, 2012)	2012	BSO για ελαχιστοποίηση του κόστους παραγωγής ηλεκτρικής ενέργειας	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την ελαχιστοποίηση του κόστους παραγωγής ηλεκτρικής ενέργειας
5	(J. Xue, Wu, Shi, & Cheng, 2012)	2012	Brain Storm Optimization Algorithm for Multi-objective Optimization Problems (MOBSO)	<ul style="list-style-type: none"> <li>• Γκαουσιανή μετάλλαξη, μετάλλαξη Cauchy για παραγωγή γενεών</li> <li>• Ομαδοποίηση με ελιτισμό</li> </ul>



6	(Ramanand, Krishnanand, Panigrahi, & Mallick, 2012)	2012	Teaching–Learning BSO (TLBSO) Algorithm Εφαρμογή σε συστήματα ηλεκτρικής ενέργειας	Απλή εφαρμογή αλγορίθμου σε συστήματα ηλεκτρικής ενέργειας
7	(Duan, Li, & Shi, 2013)	2013	Predator–Prey BSO (PPBSO) Algorithm Optimization of efficiency of the brushless DC wheel motor	<ul style="list-style-type: none"> <li>• Διαφοροποίηση στην ανανέωση και παραγωγή ιδεών</li> </ul>
8	(C. Sun, Duan, & Shi, 2013)	2013	Optimal Satellite Formation Reconfiguration Based on Closed-Loop Brain Storm Optimization (CLBSO)	<ul style="list-style-type: none"> <li>• Δημιουργία συντελεστών με closed-loop strategies</li> </ul>
9	(Zhan et al., 2013)	2013	Εξερεύνηση παραμέτρων του BSO	<ul style="list-style-type: none"> <li>• <math>p_{replace}, p_{one} \in [0, 1.0]</math></li> <li>• Δοκιμές διαφόρων τιμών κέντρων ομάδων</li> </ul>
10	(Cheng, Shi, Qin, & Gao, 2013)	2013	Ανάλυση λύσεων ομαδοποίησης του BSO	<ul style="list-style-type: none"> <li>• Γκαουσιανή κατανομή για εξερεύνηση της παραμέτρου του βήματος</li> </ul>
11	(Xie & Wu, 2014)	2014	Modified Multiobjective Brain Storm Optimization (MMBSO) Algorithm	<ul style="list-style-type: none"> <li>• Clustering strategy in the objective space instead of in the solution space</li> <li>• Potential Pareto-dominance areas στις επαναλήψεις</li> <li>• Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) clustering</li> <li>• Differential Evolution (DE)</li> </ul>
12	(X. Guo, Wu, & Xie, 2014)	2014	Self- Adaptive BSO (SBSO) for multimodal optimization	<ul style="list-style-type: none"> <li>• Max-fitness grouping cluster για υποδιαίρεση ιδεών σε υποομάδες</li> </ul>
13	(H. J. Zhou, Jiang, & Ben, 2014)	2014	Niche Brain Storm Optimization Algorithm (NBSO) for Multi-peak Function Optimization	<ul style="list-style-type: none"> <li>• Niche technology</li> </ul>

14	(Cheng, Shi, Qin, Ting, & Bai, 2014)	2014	Διατήρηση ποικιλομορφίας πληθυσμού στον BSO	<ul style="list-style-type: none"> <li>Μερική επαναρχικοποίηση λύσεων για παραγωγή ποικιλομορφίας πληθυσμού</li> </ul>
15	(Yanqiu Sun, 2014)	2014	Hybrid Approach by Integrating BSO with Grey Neural Network (BSO-GNN) πρόβλεψη χρηματιστηριακού δείκτη μετοχών	<ul style="list-style-type: none"> <li>Grey Neural Network με αλγόριθμο BSO</li> </ul>
16	(G.-W. Zhang, Zhan, Du, & Chen, 2014)	2014	Normalization Group Brain Storm Optimization Algorithm (NGBSO)/ Optimize the design of buck regulator in Power Electronic Circuit	<ul style="list-style-type: none"> <li>Normalization Group Strategy (NGS)</li> </ul>
17	(Arsuaga-Rios & Vega-Rodriguez, 2014)	2014	Multiobjective Brain Storm Algorithm σε Grid System(MOBSA)	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί τη βελτιστοποίηση κατανάλωσης ενέργειας και χρόνου
18	(J. Chen, Xie, & Ni, 2014) εξηγεί καλά τύπους	2014	Modified BSO based on uncertainty information of solution sets/ απλή αριθμητική εφαρμογή αλγορίθμου	<ul style="list-style-type: none"> <li>Αυτοπροσδιορισμός αριθμού ομάδων</li> <li>Affinity propagation (AP)</li> <li>Δημιουργία συντελεστών με Cloud Drops Algorithm</li> <li>Προσαρμοστική δημιουργία ατόμων επόμενης γενιάς</li> </ul>
19	(Duan & Li, 2015)	2015	Quantum-behaved BSO (QBSO) / Πρόβλημα πλανώδιου πωλητή με μαγνητικό πεδίο	<ul style="list-style-type: none"> <li>Βελτίωση και αύξηση ποικιλομορφίας πληθυσμού</li> <li>Χρήση global information για δημιουργία ατόμων επόμενων γενεών</li> </ul>
20	(J. Li & Duan, 2015)	2015	Εφαρμογή BSO σε συστήματα προσγείωσης F/A-18	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί τη βελτιστοποίηση σε αυτόματα συστήματα προσγείωσης F/A-18
21	(Jordehi, 2015)	2015	Ρύθμιση συσκευών FACTS <sup>3</sup> σε συστήματα ηλεκτρικής	Εφαρμογή αλγορίθμου που εξυπηρετεί τη ρύθμιση συσκευών

<sup>3</sup>FACTS= Flexible AC transmission systems (multiobjective problem)

			ενέργειας	FACTS σε συστήματα ηλεκτρικής ενέργειας
22	(Qiu & Duan, 2014)	2015	Βελτιστοποίηση παραμέτρων με τον BSO εφαρμογή σε μη επανδρωμένα εναέρια οχήματα	Απλή εφαρμογή αλγορίθμου για βελτιστοποίηση μη επανδρωμένων εναέριων οχημάτων
23	(Maftciu-Scai, 2015)	2015	Επίλυση συστήματος εξισώσεων με προσεγγιστικές λύσεις	Απλή εφαρμογή αλγορίθμου για επίλυση συστήματος εξισώσεων
24	(Huang, Wu, & Chen, 2015)	2015	Design Swapping Encryption System for Video Streaming / minimization of the standard deviation of the key space's spectrum	Απλή εφαρμογή αλγορίθμου σε συστήματα κρυπτογράφησης
25	(Cong Li & Duan, 2015)	2015	ChaoticBSO για συγχώνευση εικόνας	<ul style="list-style-type: none"> <li>• Διαφορικές εξισώσεις</li> <li>• Αρχικοποίηση ατόμων με χρήση ChaoticAlgorithm</li> </ul>
26	(Shi, 2015)	2015	Brain Storm Optimization Algorithm in Objective Space (BSO-OS algorithm)	<ul style="list-style-type: none"> <li>• Convergent operation in the 1-dimensional objective space instead of in the solution space</li> </ul>
27	(Z. Yang & Shi, 2015)	2015	BSO with chaotic operation (BSOCO)	<ul style="list-style-type: none"> <li>• BSO με chaotic operation</li> </ul>
28	(Zhu & Shi, 2015)	2015	Modified BSO algorithm	<ul style="list-style-type: none"> <li>• k-medians ομαδοποίηση</li> <li>• Αντικατάσταση κέντρου ομάδας με ένα άτομο πολύ κοντά σε αυτό</li> </ul>
29	(X. Guo, Wu, Xie, Cheng, & Xin, 2015)	2015	Self-adaptive Multiobjective Brain Storm Optimization (SMOBSO)	<ul style="list-style-type: none"> <li>• Απλή ομαδοποίηση ιδεών για μεγαλύτερη ταχύτητα</li> <li>• Προσαρμοστική μέθοδος μετάλλαξης</li> <li>• Open state probability (OSP)</li> </ul>
30	(Z. Cao, Shi, et al., 2015)	2015	Random Grouping BSO algorithm (RGBSO)	<ul style="list-style-type: none"> <li>• Δυναμική ρύθμιση παραμέτρου του βήματος</li> <li>• Αντικατάσταση μεθόδου ομαδοποίησης με στρατηγική</li> </ul>

				τυχαία ομαδοποίησης
31	(Z. Cao, Hei, Wang, Shi, & Rong, 2015)	2015	BSO with Differential Evolution Strategy for Applications of Artificial Neural Network (ANNs)	<ul style="list-style-type: none"> <li>Differential Evolution για δημιουργία ιδεών</li> </ul>
32	(Jiang, Qin, Wu, & Sun, 2015)	2015	Ανάλυση και πρόβλεψη χρονοσειρών για τις ταχύτητες ανέμου με Support Vector Regression (SVR)	Απλή εφαρμογή αλγορίθμου για ανάλυση και πρόβλεψη ανέμου
33	(Y. Yang, Shi, & Xia, 2015)	2015	Advanced discussion mechanism-based brain storm optimization (ADMBSO)	<ul style="list-style-type: none"> <li>Differential step strategy into the workflow</li> </ul>
34	(Arsuaga-Rios & Vega-Rodriguez, 2015)	2015	Energy consumption and execution time using Multi-objective BSO (MOBSA) in grid systems	Εφαρμογή αλγορίθμου σε συστήματα Grid
35	(Wei & Shi, 2015)	2015	Εφαρμογή BSO για βελτιστοποίηση ασύρματα δίκτυα αισθητήρων	Απλή εφαρμογή αλγορίθμου σε ασύρματα δίκτυα αισθητήρων
36	(Roy & Anuradha, 2015)	2015	Ομαδοποίηση με hardc-means	<ul style="list-style-type: none"> <li>Δημιουργία τυχαίων λύσεων με παραγωγή κεντροειδών</li> <li>Αξιολόγηση με μέτρα όπως Davies Boudlin Index και Dunn Index</li> </ul>
37	(J. Chen, Cheng, Chen, Xie, & Shi, 2015)	2015	Modified BSO/ Optimization of the dynamic deployments of two different wireless sensor networks (WSN)	<ul style="list-style-type: none"> <li>Affinity Propagation (AP)</li> </ul>
38	(Hua, Chen, & Xie, 2016)	2016	Πρόβλημα πλανόδιου πωλητή	<ul style="list-style-type: none"> <li>Discrete Particle Swarm Optimization Algorithm (DPSO)</li> </ul>
39	(Gupta, Swarnalatha, & Chharia, 2016)	2016	Fuzzy C-Means and BSO Ελαχιστοποίηση αποστάσεων μεταξύ των ομάδων	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την ελαχιστοποίηση αποστάσεων μεταξύ ομάδων
40	(J. Chen, Wang, Cheng, & Shi, 2016)	2016	Brain Storm Optimization with Agglomerative Hierarchical Clustering	<ul style="list-style-type: none"> <li>Agglomerative Hierarchical Clustering (AHC)</li> </ul>

			Analysis	
41	(El-Abd, 2016a)	2016	Modified BSO	<ul style="list-style-type: none"> <li>Μηχανισμός επαναρχικοποίησης μεταβλητών βάσει της τρέχουσας κατάστασης του πληθυσμού</li> <li>Ρύθμιση παραμέτρου του βήματος ανάλογα με το μέγεθος του χώρου αναζήτησης</li> </ul>
42	(Zheng, Lei, Gong, & Tang, 2016)	2016	Multi-factorial brain storm optimization algorithm (MFBSA)	<ul style="list-style-type: none"> <li>Multitasking ομαδοποίηση</li> </ul>
43	(Xinyuan Ma & Xian, 2016)	2016	A Re-estimation BSO to Train Hidden Markov Model for Transcription Factor Binding Site Analysis	Εφαρμογή αλγορίθμου που εξυπηρετεί την εκπαίδευση Μαρκοβιανού Μοντέλου
44	(L. Li, Zhang, Chu, & Niu, 2016)	2016	BSO-Full Connected , BSO-Ring connected, BSO-Star connected Algorithm	<ul style="list-style-type: none"> <li>Topology structure (full connected, ring connected and star connected )concept in BSO</li> </ul>
45	(Sai et al., 2016)	2016	Μελέτη και σύγκριση πρόσφατων ευριστικών μεθόδων βελτιστοποίησης	Στη μελέτη αυτή εντοπίζονται και συγκρίνονται πρόσφατες ευριστικές μέθοδοι βελτιστοποίησης
46	(Xueming Yan, Hao, Huang, & Li, 2016)	2016	BSO with human intelligence in computer game/ εφαρμογή αλγορίθμου στο πρόβλημα δρομολόγησης οχημάτων	Απλή εφαρμογή αλγορίθμου στο πρόβλημα δρομολόγησης οχημάτων
47	(Y. Xue, Tang, & Ma, 2016)	2016	Εύρεση της βέλτιστης λύσης μαθηματικού μοντέλου	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί στην εύρεση της βέλτιστης λύσης μαθηματικού μοντέλου
48	(Cervantes-Castillo & Mezura-Montes, 2016)	2016	A study of constraint-handling techniques in Brain Storm Optimization Algorithm (BSO), Modified Brain Storm Optimization Algorithm (MBSO) and Simple Modified Brain Storm Optimization Algorithm (SMBSO)	Στη μελέτη αυτή εντοπίζονται και συγκρίνονται τεχνικές από τρία διαφορετικά είδη του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών

49	(Jianzhou Wang, Hou, Wang, & Shen, 2016)	2016	A hybrid v-SVR model, with principal component analysis (PCA) and BSO πρόβλεψη χρηματιστηριακού δείκτη μετοχών	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την παρακολούθηση και πρόβλεψη ενός χρηματιστηριακού δείκτη μετοχών
50	(Soyinka & Duan, 2016)	2016	Ελαχιστοποίηση κατανάλωσης καυσίμων στους δορυφόρους	<ul style="list-style-type: none"> <li>• Lambert formulation</li> </ul>
51	(Cheng, Qin, Chen, Wang, & Shi, 2016)	2016	Brain Storm Optimization in Objective Space Algorithm (BSO-OS) for Multimodal Optimization Problems	Απλή αριθμητική εφαρμογή αλγορίθμου για πολυτροπικά προβλήματα
52	(Niu, Liu, Liu, & Yang, 2016)	2016	Simplified BSO (SBSO), BSO in objective space (BSO-OS) πρόβλημα βελτιστοποίησης χαρτοφυλακίου	Απλή εφαρμογή αλγορίθμου στο πρόβλημα βελτιστοποίησης χαρτοφυλακίου
53	(Partha S Pal, Kar, Mandal, & Ghoshal, 2016)	2016	Identification of NARMAX Hammerstein <sup>4</sup> models with performance assessment	Απλή εφαρμογή αλγορίθμου σε μοντέλα NARMAX Hammerstein
54	(Y. Wu, Xie, & Liu, 2016)	2016	Finding the knee point of Pareto-optimal front	<ul style="list-style-type: none"> <li>• Clustering strategy directly in the objective space instead of in the solution space</li> <li>• Αξιολόγηση knee regions</li> <li>• Differential Evolution (DE)</li> </ul>
55	(Ting & Shi, 2016)	2016	Εκτίμηση παραμέτρου σε two-soil layer model	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την εκτίμηση παραμέτρου
56	(Zhu & Shi, 2016)	2016	Επίλυση προβλημάτων κάλυψης των ασύρματων δικτύων αισθητήρων σε διάφορες περιοχές	<ul style="list-style-type: none"> <li>• Grid based strategy</li> </ul>
57	(Yadav, 2016)	2016	Ανάκτηση και επεξεργασία ιατρικών δεδομένων	<ul style="list-style-type: none"> <li>• Adaptive fractional brain storm optimization</li> <li>• Fractional calculus</li> <li>• Parametric-enabled similarity measure</li> </ul>
58	(Jiang & Li, 2016)	2016	Πρόγνωση βραχυπρόθεσμης	<ul style="list-style-type: none"> <li>• Discrete wavelet transform</li> </ul>

<sup>4</sup> Το μοντέλο Hammerstein είναι ένα από τα πιο σημαντικά μοντέλα που χρησιμοποιούνται στη μηχανική για τη μοντελοποίηση μη γραμμικών συστημάτων.

			ταχύτητας ανέμου	(DWT) • Back propagation neural network (BPNN)
59	(Cheng, Qin, Chen, & Shi, 2016)	2016	Βιβλιογραφική ανασκόπηση του αλγορίθμου BSO	Στη παρούσα δημοσίευση πραγματοποιείται βιβλιογραφική ανασκόπηση του αλγορίθμου
60	(Z. Zhou, Duan, & Shi, 2016)	2016	Ανάλυση σύγκλισης του BSO	• Χρήση μαρκοβιανού μοντέλου
61	(Madheswari, Venkateswaran, & Sowmiya, 2016)	2016	Βελτιστοποίηση συγχώνευσης εικόνας με χρήση του BSO	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί τη βελτιστοποίηση συγχώνευσης εικόνας
62	(El-Abd, 2016b)	2016	Cooperative Co-evolutionary BSO (CCBSO) algorithm	Απλή αριθμητική εφαρμογή αλγορίθμου
63	(Ramadan & Khedr, 2016)	2017	BSO για επίλυση προβλήματος συνδεσιμότητας στο δίκτυο ασύρματων αισθητήρων	Απλή εφαρμογή αλγορίθμου για επίλυση προβλήματος συνδεσιμότητας στο δίκτυο ασύρματων αισθητήρων
64	(G.-G. Wang, Hao, Cheng, Shi, & Cui, 2017)	2017	BSO και θεωρία γράφων	• Τα άτομα κάθε γενιάς μεταφέρονται σε έναν μη κατευθυνόμενο γράφο με βάρος (undirectedweightgraph (UWG)) • Αξιολόγηση κατάσταση σύγκλισης με Χαμιλτονιανό κύκλο
65	(Partha Sarathi Pal, Kar, Mandal, & Ghoshal, 2017)	2017	Parametric Identification with Performance Assessment of Wiener Systems using BSO	Απλή εφαρμογή αλγορίθμου σε συστήματα Wiener
66	(El-Abd, 2017)	2017	Global-best BSO (GBSO)	• επαναρχικοποίηση μεταβλητών βάσει της τρέχουσας κατάστασης του πληθυσμού
67	(X.-L. Wu, Zhang, & Li, 2017)	2017	(BSO) algorithm integrating the population diversity and discussion mechanism (PD-DMBSO) flexible job shop scheduling problem (FJSP)	• Perturbation operator and a combining operator για παραγωγή νέων ατόμων
68	(Garcia-Ródenas, Linares, & López-	2017	A cooperative Brain Storm Optimization Algorithm	Απλή αριθμητική εφαρμογή αλγορίθμου

	Gómez, 2017)			
69	(H. Wang, Liu, Yi, Niu, & Baek, 2017)	2017	BSO with learning strategy (BSOLS)	<ul style="list-style-type: none"> <li>• Learning strategy</li> </ul> <p>Διατήρηση των ατόμων με την καλύτερη συνάρτηση καταλληλότητας</p>
70	(Jin & Qin, 2017)	2017	A GPU-based implementation of BSO	<ul style="list-style-type: none"> <li>• Χρήση NVIDIA's CUDA τεχνολογία</li> </ul>
71	(Chandrasekar & Khare, 2017)	2017	BSFS: Design and Development of Exponential Brain Storm Fuzzy System Ταξινόμηση δεδομένων	<ul style="list-style-type: none"> <li>• BSO με fuzzy συστήματα</li> <li>• Εκθετικό μοντέλο</li> </ul>
72	(Z. Cao, Rong, & Du, 2017)	2017	BSO algorithm with dynamic clustering strategy (BSO-DCS)	<ul style="list-style-type: none"> <li>• Dynamic clustering strategy (DCS)</li> </ul>
73	(Cheng et al., 2017)	2017	Έρευνα του BSO	Στην έρευνα αυτή εντοπίζονται και καταγράφονται οι τελευταίες εφαρμογές του BSO
74	(Verma & Dubey, 2017)	2017	Fuzzy BSO and Euclidean distance (EED) for the vein based recognition system	<ul style="list-style-type: none"> <li>• Fuzzy BSO</li> <li>• Ευκλείδεια απόσταση</li> </ul>
75	(Y. Cao et al., 2017)	2017	Analyzing numerical effects of different components of BSO via visualizing confidence intervals	Σε αυτή τη δημοσίευση πραγματοποιείται ανάλυση αριθμητικών δεδομένων διαφόρων παραμέτρων του BSO
76	(Y. Wu, Wang, Fu, & Xu, 2017)	2017	Difference Brain Storm Optimization (DBSO) to solve combined heat and power economic dispatch (CHPED) problem in power plant	<ul style="list-style-type: none"> <li>• DE mutation</li> </ul>
77	(X. Chu, Chen, Cai, Chen, & Niu, 2017)	2017	Augmented brain storm optimization (ABSOS) with mutation strategies	<ul style="list-style-type: none"> <li>• Non-uniform mutation</li> <li>• DE random mutation</li> </ul>
78	(Yu et al., 2017)	2017	Brain storm optimization with adaptive search radius for optimization	<ul style="list-style-type: none"> <li>• Adaptive search radius method</li> </ul>
79	(Tuba, Dolicanin, & Tuba, 2017)	2017	BSO with chaotic local search (CBSO)	<ul style="list-style-type: none"> <li>• BSO with chaotic local search (CLS)</li> </ul>
80	(Tuba, Dolicanin,	2017	Brain storm optimization	<ul style="list-style-type: none"> <li>• Chaotic maps</li> </ul>



	et al., 2017)		algorithm with chaotic maps	
81	(Qiu, Duan, Zhou, Hu, & Shi, 2017)	2017	Chaotic predator-prey brain storm optimization	<ul style="list-style-type: none"> <li>• Predator-prey operation</li> <li>• Chaotic operation</li> </ul>
82	(Pérez-Cruz, Portilla-Flores, Niño-Suárez, & Rivera-Blas, 2017)	2017	Minimization of quadratic performance index in chaotic systems	Απλή εφαρμογή αλγορίθμου σε chaotic συστήματα
83	(Anbarasi & Saleem Durai, 2018)	2018	Modified Brainstorm Optimization Algorithm/ prediction of protein folding kinetics	<ul style="list-style-type: none"> <li>• Fuzzy C-Means (FCM)</li> </ul>
84	(Ganesh & Kanimozhi, 2018)	2018	Προσέγγιση για τοποθέτηση πυκνωτή με βάση το δείκτη εξισορρόπησης φορτίου στο ακτινικό σύστημα διανομής (radialdistributionsystem)	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την προσέγγιση για τοποθέτηση πυκνωτή με βάση το δείκτη εξισορρόπησης φορτίου στο ακτινικό σύστημα διανομής
85	(Fu, Wang, & Huang, 2018)	2018	Integrated scheduling for a distributed manufacturing system/ stochastic multi-objective model	Απλή εφαρμογή αλγορίθμου
86	(Qiao, Huang, & Gao, 2018)	2018	Ανάλυση ασυμπτωτικής σύγκλισης BSO	Σε αυτή την έρευνα μελετάται και αναλύεται η ασυμπτωτική σύγκλιση του αλγορίθμου
87	(W. Chen et al., 2018)	2018	Simplex search-based BSO (Simplex-BSO)	<ul style="list-style-type: none"> <li>• the Nelder-Mead Simplex method</li> </ul>
88	(Yuehong Sun, Jin, & Wang, 2018)	2018	BSO with multi-population based ensemble of creating operations (MPEBSO)	<ul style="list-style-type: none"> <li>• Differential Evolution</li> <li>• Προσθήκη θορύβου για δημιουργία ατόμων</li> </ul>
89	(Chunquan Li et al., 2018)	2018	Vector Grouping Learning BSO for Global Optimization Problems	<ul style="list-style-type: none"> <li>• Vector grouping learning (IC-VGL) scheme</li> <li>• Hybrid individuals' update (H-IU) scheme</li> <li>• Random grouping (RG) scheme</li> </ul>
90	(X. Zhang, Zhang, Gao, & He, 2018)	2018	BSO-based feature selection algorithm (CBSOFS)	<ul style="list-style-type: none"> <li>• Selective probability-based real encoding strategy</li> </ul>
91	(Dolicinanin,	2018	Προγραμματισμός διαδρομών	Απλή εφαρμογή αλγορίθμου που

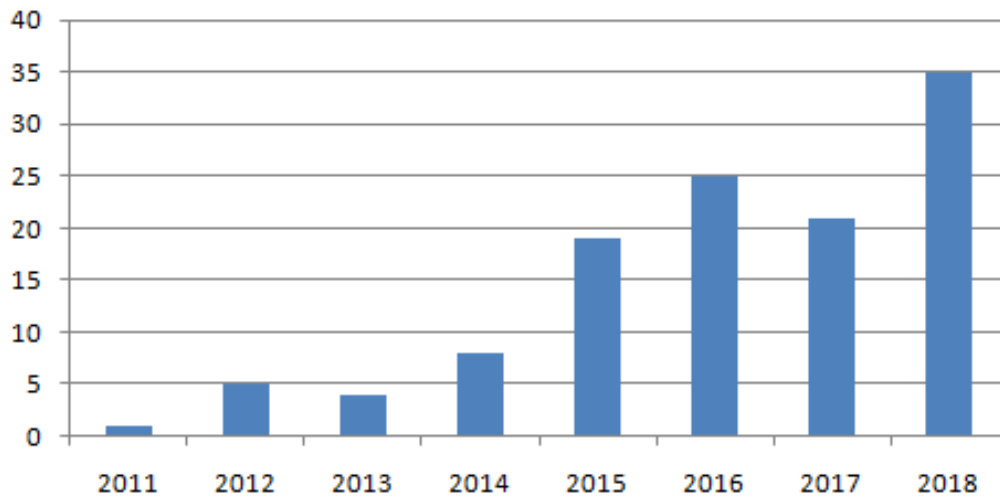
	Fetahovic, Tuba, Capor-Hrosik, & Tuba, 2018)		μη επανδρωμένων οχημάτων εναέριας κυκλοφορίας	αποσκοπεί στο σχεδιασμό και προγραμματισμό διαδρομών μη επανδρωμένων οχημάτων εναέριας κυκλοφορίας
92	(Cheng, Lu, Song, Chen, & Shi, 2018)	2018	BSO for dynamic multimodal optimization problem	Απλή αριθμητική εφαρμογή αλγορίθμου
93	(Margain, Ochoa, Almaguer, & Velázquez, 2017)	2018	Model on Oil Platform Using BSO	Απλή εφαρμογή αλγορίθμου
94	(Tuba, Capor-Hrosik, Alihodzic, & Tuba, 2017)	2018	Εύρεση βέλτιστης θέσης πολυκοπτέρων (drones) με ελάχιστη κατανάλωση ενέργειας	Απλή εφαρμογή αλγορίθμου που αποσκοπεί στην εύρεση βέλτιστης θέσης πολυκοπτέρων (drones) με ελάχιστη κατανάλωση ενέργειας
95	(Yadav, 2018)	2018	Fractional brain storm optimization (FBSO) ανάκτηση ιατρικών εικόνων	<ul style="list-style-type: none"> <li>• Fractional Lion Algorithm (FLA)</li> </ul>
96	(Cervantes-Castillo, Leguizamon, & Mezura-Montes, 2017)	2018	Modified BSO and a boundary search operator (MBSO-BS) διερεύνηση ορίων	<ul style="list-style-type: none"> <li>• Τελεστής αναζήτησης ορίων</li> </ul>
97	(Revathi, Ramaraj, & Chithra, n.d.)	2018	Brainstorm-based Whale Optimization Algorithm/ προστασία ιδιωτικών δεδομένων που δημοσιεύονται σε cloud computing	<ul style="list-style-type: none"> <li>• WhaleOptimization Algorithm</li> </ul>
98	(Ravi & Khare, 2018)	2018	Brain genetic fuzzy system (BGFS) ταξινόμηση δεδομένων	<ul style="list-style-type: none"> <li>• Exponential Genetic BSO</li> </ul>
99	(Xuesong Yan, Zhu, Wu, Gong, & Wang, 2018)	2018	BSO for AVO elastic parameter inversion problem	Απλή εφαρμογή του αλγορίθμου που εξυπηρετεί τη ρύθμιση παραμέτρων
100	(Xu, Wu, Fu, Wang, & Lu, 2018)	2018	Discrete BSO algorithm Πρόβλημα πλανόδιου πωλητή	<ul style="list-style-type: none"> <li>• Objective space instead of the solution space</li> </ul>
101	(Ibrahim, Elaziz, Ewees, Selim, &	2018	Galaxy images classification	<ul style="list-style-type: none"> <li>• Moth Flame Optimization (MFO)</li> </ul>

	Lu, 2018)			
102	(Yu, Gao, Wang, Cheng, & Todo, 2018)	2018	Adaptive step length based on memory selection BSO (ASBSO)	<ul style="list-style-type: none"> <li>• Multiple step lengths</li> <li>• Memory storing mechanism (IMS)</li> </ul>
103	(Hong, Hao, Mai, Huang, & Kumar Sangaiah, 2018)	2018	K2 with brain storm optimization (K2-BSO)	<ul style="list-style-type: none"> <li>• Αλγόριθμος K2</li> </ul>
104	(Afonso, Passos, & PauloPapa, 2018)	2018	Modified BSO with OPF algorithm	<ul style="list-style-type: none"> <li>• Optimum-Path Forest (OPF) algorithm</li> </ul>
105	(Tuba, Simian, Dolicanin, Jovanovic, & Tuba, 2018)	2018	BSO for placement of multiple sinks in a network consisting of regular sensors and gateways with higher battery capacity	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί τη σωστή τοποθέτηση αισθητήρων
106	(Liang et al., 2018)	2018	Multi-objective Brainstorm Optimization Algorithm for Sparse Optimization	<ul style="list-style-type: none"> <li>• Iterative half-thresholding operator in the frame of L1/2 regulation</li> </ul>
107	(Tuba, Strumberger, Zivkovic, Bacanin, & Tuba, 2018)	2018	Mobile Robot Path Planning	<ul style="list-style-type: none"> <li>• Each new candidate solution moves to the local best position</li> </ul>
108	(Xiong, Shi, Zhang, & Zhang, 2018)	2018	Binary coded BSO (BCBSO)/ Fault section diagnosis of power systems	<ul style="list-style-type: none"> <li>• Χρήση δυαδικών διανυσμάτων</li> </ul>
109	(Sato & Fukuyama, 2018)	2018	Total Optimization of Smart City by BSO	<ul style="list-style-type: none"> <li>• Differential Evolution Strategies</li> </ul>
110	(Q. Guo et al., 2018)	2018	Πρόβλεψη ισχύος μονάδας ανεμοστρόβιλων	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί την πρόβλεψη ισχύος μονάδας ανεμοστρόβιλων
111	(Maftiu-Scai, Maftiu-Scai, & Voinea, 2017)	2018	Μείωση του εύρους ζώνης των αραιών μητρών	Απλή εφαρμογή αλγορίθμου που εξυπηρετεί στη μείωση του εύρους ζώνης των αραιών μητρών
112	(Nucamendi-Guillén, Dávila, Camacho-Vallejo,	2018	Sales territory design problem	<ul style="list-style-type: none"> <li>• Διακριτοί μηχανισμοί για παραγωγή ιδεών</li> <li>• Ελιτισμός</li> </ul>

	& González-Ramírez, 2018)			
113	(Jingyu Wang, Chen, Xue, & Du, 2018)	2018	Διερεύνηση αλγορίθμου BSO	Μελέτη που εξυπηρετεί στη διερεύνηση του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών
114	(Duan, Zhang, Shi, & Deng, 2018)	2018	Metric-distance brain storm optimization (MDBSO)/ optimization of benchmark controller	<ul style="list-style-type: none"> <li>• Ομαδοποίηση με μετρικές αποστάσεις</li> <li>• Κατανομή Lévy</li> </ul>
115	(Ke, 2018)	2018	Πρόβλημα δρομολόγησης οχημάτων	<ul style="list-style-type: none"> <li>• Convergent and divergent operations based on cumulative capacitated vehicle routing problem</li> </ul>
116	(Dash, Joshi, & Trivedi, 2018)	2018	Improved brain storm optimization (IMBSO) multiobjective analog/RF circuit sizing	<ul style="list-style-type: none"> <li>• k-means ++</li> <li>• Random probabilistic decision-making of river formation dynamics scheme</li> <li>• Αυτόματα προσαρμοζόμενη μετάλλαξη</li> </ul>

Πίνακας 6 Εφαρμογές BSO (2011-2017)

Στην **Εικόνα 10**, παρουσιάζεται το πλήθος δημοσιεύσεων του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών, κατά τη διάρκεια των ετών 2011 έως 2018, στις βάσεις δεδομένων Scopus και Google Scholar, που πραγματοποιήθηκε το Δεκέμβριο του 2018. Παρατηρείται μεγάλη αύξηση των δημοσιεύσεων, ειδικότερα κατά το έτος 2018, όπου ο αριθμός ανέρχεται στις περίπου 34 δημοσιεύσεις. Όπως είναι φανερό και από το παραπάνω διάγραμμα, ο συγκεκριμένος αλγόριθμος είναι σχετικά πρόσφατος, καθώς δημιουργήθηκε το 2011 και στατιστικά υπάρχει αυξανόμενο ενδιαφέρον της μελέτης και εφαρμογής του. Συνολικά βρέθηκαν και καταγράφηκαν 116 δημοσιεύσεις.



**Εικόνα 10** Πλήθος δημοσιεύσεων Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών. Πηγή: Scopus, Google Scholar

## 5.Καταιγισμός Ιδεών για τη Βελτιστοποίηση Παραμέτρων Μηχανών Διανυσμάτων Στήριξης

Οι Μηχανές Διανυσμάτων Στήριξης(ή μηχανές διανυσματικής υποστήριξης)(*Support Vector Machines(SVM)*) προτάθηκαν, όπως λέγεται (Burges, 1998a; Vapnik, 1995), περίπου το 1979, από τον Vapnik κι έχουν ως στόχο την εύρεση της υπερεπιφάνειας που θα διαχωρίζει τις κλάσεις με τέτοιο τρόπο ώστε το περιθώριο (*margin*)μεταξύ τους να είναι το μέγιστο δυνατό. Ο υπολογισμός του περιθωρίου προκύπτει από τη διαφορά της απόστασης του υπερεπιπέδου  $g_2$  από την αρχή των αξόνων μείον την απόσταση του υπερεπιπέδου  $g_1$  από την αρχή των αξόνων (**Εικόνα 11**). Παρακάτω παρουσιάζονται οι αντίστοιχοι τύποι για τον υπολογισμό (Burges, 1998b).

$$g(x, w) = wx^T + w_o = 0 \rightarrow z = \frac{|w_o|}{\|w\|} \quad (5.1)$$

$$g_1(x, w) = wx^T + w_o - 1 = 0 \rightarrow z_1 = \frac{|w_o+1|}{\|w\|} \quad (5.2)$$

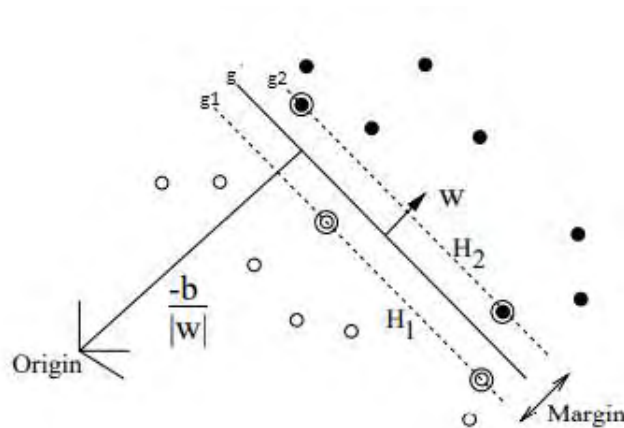
$$g_2(x, w) = wx^T + w_o + 1 = 0 \rightarrow z_2 = \frac{|w_o-1|}{\|w\|} \quad (5.3)$$

Χωρίς βλάβη της γενικότητας θεωρείται αυθαίρετα πως η κοντινότερη απόσταση κάθε επιπέδου από την ευθεία διαχωρισμού είναι 1. Επομένως το περιθώριο  $\Pi$ , δίνεται από τον παρακάτω τύπο:

$$\Pi = z_2 - z_1 = \frac{2}{\|w\|} \quad (5.4)$$

Η ποσότητα  $\frac{2}{\|w\|}$  ονομάζεται **απόσταση διαχωρισμού**. Είναι προφανές πως το βέλτιστο υπερεπίπεδο είναι εκείνο που μεγιστοποιεί όσο το δυνατόν περισσότερο την απόσταση διαχωρισμού. Όσο μεγαλύτερη είναι η απόσταση αυτή, τόσο μεγαλύτερη είναι και η ικανότητα για γενίκευση.

## 5.1 Γραμμική Κατηγοριοποίηση



Εικόνα 11 Γραμμικά Διαχωρίσιμη Κατηγοριοποίηση (Burges, 1998b)

### 5.1.1 Γραμμικά Διαχωρίσιμη Κατηγοριοποίηση

Έστω ένα διάνυσμα  $x_i \in R^n, \forall i = 1, 2, \dots, N$  και έστω ότι υπάρχει ένα υπερεπίπεδο που διαχωρίζει τις θετικές από τις αρνητικές κλάσεις. Τότε το διάνυσμα αυτό θα πρέπει να βρίσκεται από την εξωτερική πλευρά του περιθωρίου (Burges, 1998b). Για την περίπτωση των γραμμικά διαχωρίσιμων κλάσεων, ο αλγόριθμος ψάχνει για το διαχωρίσιμο υπερεπίπεδο με το ευρύτερο περιθώριο. Το παραπάνω επιτυγχάνεται αν όλα τα δεδομένα εκπαίδευσης (*training data*) ικανοποιούν τις παρακάτω σχέσεις:

$$wx_i^T + w_o \geq 1, \text{ αν } d_i = +1 \quad (5.5)$$

$$wx_i^T + w_o \leq -1, \text{ αν } d_i = -1 \quad (5.6)$$

Οι δύο αυτές σχέσεις ισοδύναμα σε μία γράφονται ως εξής:

$$d_i(wx_i^T + w_o) - 1 \geq 0, \forall i \quad (5.7)$$

Για την εύρεση του βέλτιστου υπερεπιπέδου σύμφωνα με τον περιορισμό  $d_i(wx_i^T + w_o) \geq 1, \forall i = 1, 2, \dots, N$  στόχος είναι η μεγιστοποίηση της απόστασης διαχωρισμού  $\Delta = \Delta_{max} = \frac{2}{\|w\|}$  κι αυτό επιτυγχάνεται με την εύρεση των βέλτιστων τιμών  $w_o, w$ . Το διάνυσμα βάρους  $w$  ελαχιστοποιεί τη συνάρτηση κόστους  $J(w) = \frac{1}{2} w^T w$ . Η συνάρτηση παίρνει αυτή τη μορφή διότι έτσι διευκολύνεται η παραγωγή και η εύρεση ελαχίστου. Επιπλέον, δεδομένου ότι η συνάρτηση αυτή είναι κυρτή, η ελαχιστοποίηση με γραμμικούς περιορισμούς μπορεί να επιτευχθεί με τη χρήση πολλαπλασιαστών Lagrange. Το πρόβλημα, λοιπόν, ανάγεται σε πρόβλημα τετραγωνικού προγραμματισμού (*quadratic programming*).

Έστω ότι στόχος είναι η ελαχιστοποίηση μίας τετραγωνικής αντικειμενικής συνάρτησης (*objective function*)  $J(w)$  υπό τους περιορισμούς  $g(x_p, w) \geq 0, \forall i = 1, 2, \dots, N$ . Η γενική μορφή της τετραγωνικής συνάρτησης είναι  $J(x) = \frac{1}{2} w^T Q_w w + a^T w$ . Η σχέση ισοδύναμα, ελαχιστοποιώντας τη

Lagrangian  $L(w, \lambda_i)$  ως προς  $w, w_o$ , χρησιμοποιώντας τη συνθήκη βελτιστοποίησης Karush-Kuhn-Tucker, δηλαδή  $\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial w_o} = 0$ , γίνεται:

$$L = J(w) - \lambda_i \sum_i g(x_i, w) \quad (5.8)$$

Οι πίνακες  $Q, a$  είναι δεδομένα και οι συντελεστές  $\lambda_i \geq 0, i = 1, 2, \dots, N$  ονομάζονται **πολλαπλασιαστές Lagrange**.

Για την προκειμένη περίπτωση η τετραγωνική συνάρτηση για  $Q = I$  και  $a = 0$  είναι

$$\frac{\sqrt{\|w\|^2}}{2} = \frac{\sqrt{ww^T}}{2} \quad (5.9)$$

η οποία ισοδύναμα γίνεται:

$$J(w) = \frac{\|w\|^2}{2} \quad (5.10)$$

με περιορισμούς  $g(x_i, w) = d_i(wx_i^T + w_o) - 1 \geq 0$ , ενώ η μεγιστοποίηση του  $\Delta = \Delta_{max} = \frac{2}{\|w\|}$  είναι η ελαχιστοποίηση του όρου  $\frac{\|w\|}{2}$ . Τελικά, προκύπτει ο τύπος

$$L = \frac{1}{2} \|w\|^2 - \sum_i \lambda_i d_i (wx_i^T, w_o) + \sum \lambda_i \quad (5.11)$$

Συμπερασματικά, η εξίσωση του υπερεπιπέδου είναι:

$$g(x, w) = 0 \leftrightarrow \sum_{i=1, i: \lambda_i > 0}^N \lambda_i x_i x_i^T + w_o = 0 \quad (5.12)$$

Η αντίστοιχη συνάρτηση διαχωρισμού είναι:

$$F(x, w) = \text{sgn}[g(x, w)] = \text{sgn}(\sum_{i=1, i: \lambda_i > 0}^N \lambda_i x_i x_i^T + w_o) \quad (5.13)$$

## 5.1.2 Γραμμικά Μη Διαχωρίσιμη Κατηγοριοποίηση

Τα γραμμικά διαχωρίσιμα μοντέλα γίνονται λιγότερο αυστηρά όταν εισάγεται η έννοια της μεταβλητής χαλάρωσης (*slack variable*)  $\xi_i \geq 0$  ούτως ώστε για να συμβεί σφάλμα να πρέπει  $\xi_i \geq 1, \forall i = 1, 2, \dots, N$  (Burges, 1998b). Επομένως, προκύπτουν οι παρακάτω αλλαγές

$$g(x, w) = wx_i^T + w_o = 0 \quad (5.14)$$

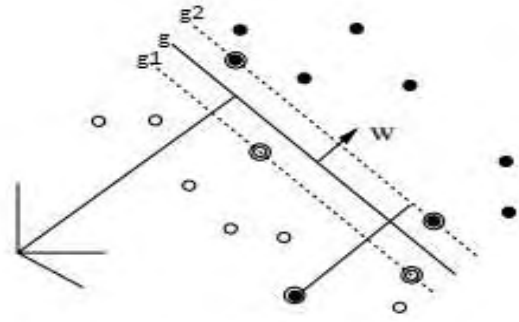
$$g_1(x, w) = wx_i^T + w_o - 1 = 0 \rightarrow wx_i^T + w_o - 1 + \xi \geq 0 \text{ αν } d_i = +1 \quad (5.15)$$

$$g_2(x, w) = wx_i^T + w_o + 1 = 0 \rightarrow wx_i^T + w_o + 1 - \xi \leq 0 \text{ αν } d_i = -1 \quad (5.16)$$

Ισοδύναμα τα παραπάνω σε μία σχέση γράφονται:

$$d_i (wx_i^T + w_o) - 1 + \xi \geq 0 \quad (5.17)$$





Εικόνα 12 Γραμμικά Μη Διαχωρίσιμη Κατηγοριοποίηση(Burges, 1998b)

Αντίστοιχα με τη γραμμικά διαχωρίσιμη κατηγοριοποίηση είναι και το παρόν πρόβλημα. Πιο συγκεκριμένα, το σύνολο των σφαλμάτων σε όλα τα σημεία θα δίνεται από τον τύπο

$$\sum_{i=1}^N \xi_i \quad (5.18)$$

και το κριτήριο που πρέπει να ελαχιστοποιηθεί είναι:

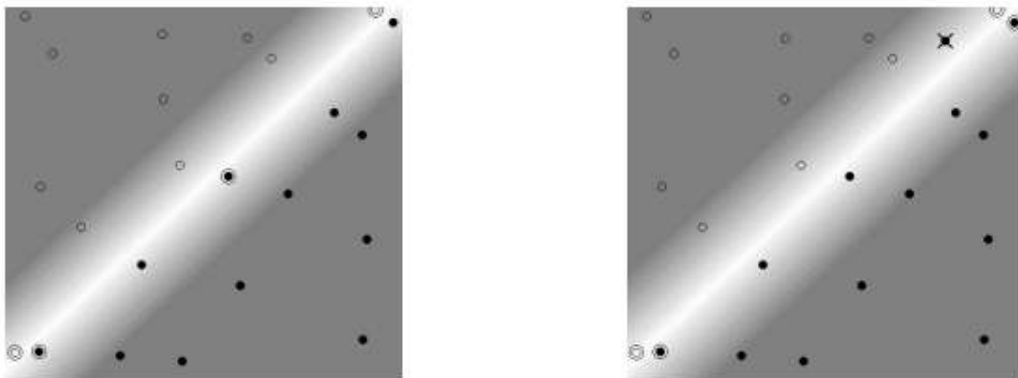
$$J(w) = \frac{ww^T}{2} + C \sum_{i=1}^N \xi_i \quad (5.19)$$

όπου  $C \geq 0$  είναι μία παράμετρος κόστους που επιλέγεται από τον ίδιο τον χρήστη. Όσο μεγαλύτερο είναι το  $C$  τόσο μεγαλύτερη είναι η ποινή που αποδίδεται στα σφάλματα.

Στη συγκεκριμένη κατηγορία ταξινόμησης, η Lagrangian θα είναι πολυπλοκότερη δεδομένου του γεγονότος ότι θα διαθέτει πολλαπλασιαστές  $\mu_i, \xi_i$ . Η τελική μορφή της Lagrangian καταλήγει να εξαρτάται μόνο από τους πολλαπλασιαστές  $\lambda_i$  που αντιστοιχούν στα σημεία  $x_i$ . Η εξίσωση του υπερπιπέδου όπως και προηγουμένως δίνεται από τον τύπο:

$$g(x, w) = 0 \leftrightarrow \sum_{i=1, i: 0 < \lambda_i \leq C}^N \lambda_i x_i x^T + w_0 = 0 \quad (5.20)$$

Η διαφορά έγκειται στο ότι τα διανύσματα  $x_i$  έχουν πολλαπλασιαστές που κυμαίνονται στο διάστημα  $(0, C]$ . Μεγάλο  $C$  μπορεί να οδηγήσει σε υπερπροσαρμογή στο δείγμα εκπαίδευσης ιδιαίτερα αν το  $N$  δεν είναι αρκετά μεγάλο και αντιπροσωπευτικό. Επίσης, μεγάλο  $C$  μπορεί να καθυστερήσει την εκπαίδευση εφόσον το εύρος των τιμών των  $\lambda_i$  που διερευνώνται για τη λύση του προβλήματος αυξάνεται.



Εικόνα 13 Απεικόνιση της γραμμικά διαχωρίσιμης (αριστερά) και μη διαχωρίσιμης (δεξιά) ταξινόμησης(Burges, 1998b)

## 5.2 Μη Γραμμική Κατηγοριοποίηση

Όλες οι παραπάνω μέθοδοι που αναφέρθηκαν, μπορούν να γενικευτούν για τη μη γραμμική κατηγοριοποίηση (*Soft Margin*). Οι (Boser, Guyon, & Vapnik, 1992) έδειξαν ότι αυτό μπορεί να επιτευχθεί αν εφαρμοστεί μία παλαιότερη μέθοδος του (Aizerman, 1964) στην οποία βάση δίνεται στο εσωτερικό γινόμενο  $\langle x_i x_j \rangle = x_i x_j^T$ . Αν τα εσωτερικά γινόμενα αντικατασταθούν από τις κατάλληλες συναρτήσεις, μπορεί να επιτευχθεί η κατασκευή μίας γραμμικής συνάρτησης διαχωρισμού με τη χρήση μίας απεικόνισης που καλείται  $\Phi$ . Έτσι, ορίζεται:

$$\Phi = R^n \rightarrow H \quad (5.21)$$

Με την αντίστοιχη αντικατάσταση προκύπτει η σχέση  $\langle \varphi(x_i) \varphi(x_j) \rangle$ . Αν υπάρχει μία συνάρτηση πυρήνα (*kernel function*)  $K$  τέτοια ώστε  $K(x_i x_j) = \varphi(x_i) \varphi(x_j)$ , θα απαιτούνταν να ήταν γνωστό μόνο το  $K$  για την εκπαίδευση του αλγορίθμου, χωρίς να χρειάζεται να είναι γνωστή η τιμή του  $\Phi$ .

Για να χρησιμοποιηθεί μία συνάρτηση ως συνάρτηση πυρήνα θα πρέπει να ικανοποιεί τη συνθήκη του Mercer.

## 5.3 Συνθήκη του Mercer

Πολλές φορές δεν είναι προφανές και σίγουρο αν υπάρχει ζευγάρι  $\{H, \Phi\}$  ή όχι με τις ιδιότητες που προαναφέρθηκαν για συγκεκριμένους πυρήνες (Burgess, 1998a). Η συνθήκη του Mercer (*Mercer's condition*) που αναπτύχθηκε από τους Vapnik, Courant και Hilbert χρησιμοποιείται για να εξετάσει το ζήτημα αυτό.

Έτσι, η συνθήκη επισημαίνει πως υπάρχει απεικόνιση  $\Phi$  και συνάρτηση πυρήνα  $K(x_i x_j) = \varphi(x_i) \varphi(x_j)$  αν και μόνο αν

$$\int z^2(x) dx \leq \infty \quad (5.22)$$

Τότε η συμμετρικά και θετικά ορισμένη συνάρτηση  $K(x, y)$  ικανοποιεί τη συνθήκη

$$\int \int K(x, y) z(x) z(y) dx dy \geq 0 \quad (5.23)$$

Θα πρέπει να αναφερθεί ότι η συνθήκη του Mercer εξετάζει πότε ένας υποψήφιος πυρήνας είναι στην πραγματικότητα εσωτερικό γινόμενο (*dot product*) ή όχι σε κάποιο χώρο, όμως δεν αναφέρεται στο πώς θα γίνει η κατασκευή της απεικόνισης  $\Phi$  ή στον προσδιορισμό του  $H$  αντίστοιχα.

## 5.4 Συναρτήσεις πυρήνα

Ο μεγαλύτερος περιορισμός των μηχανών διανυσματικής στήριξης είναι η επιλογή του πυρήνα (Burgess, 1998b). Όταν ένας πυρήνας είναι προκαθορισμένος, τότε οι ταξινομητές έχουν μόνο

τις παραμέτρους που ορίζονται από τον χρήστη για την τιμή κόστους (*error penalty*) να λάβουν υπόψη. Οι συχνότερες χρησιμοποιούμενες συναρτήσεις πυρήνα (*kernel functions*) είναι οι εξής:

- Γραμμική (*Linear*):  $K(x, y) = xy$
- Πολυωνυμική (*Polynomial*):  $K(x, y) = (xy + C_0)^p$
- Γκαουσιανή (*RBF*):  $K(x, y) = e^{-\frac{\|x-y\|^2}{\gamma}}$
- Σιγμοειδής (*Sigmoid*):  $K(x, y) = \tanh(\kappa xy - \delta)$

όπου  $p, \delta, \kappa, \gamma = 2\sigma^2$  είναι σταθερές. Ειδικότερα, η μεταβλητή  $\sigma^2$  της γκαουσιανής συνάρτησης είναι το **πλάτος**, το οποίο είναι κοινό για όλους τους πυρήνες και καθορίζεται εκ των προτέρων.

Τα SVM έχουν τόσο πλεονεκτήματα όσο και μειονεκτήματα. Αρχικά, όσον αφορά τα πλεονεκτήματα, οδηγούν σε μοναδική λύση με τη χρήση μόνο της συνάρτησης πυρήνα, καθώς αξιοποιούν διαφορετικές συναρτήσεις διαχωρισμού. Επιπρόσθετα, αντιστέκονται στην υπερπροσαρμογή, δεν επηρεάζονται εύκολα από τη διάσπαση τη χώρου των χαρακτηριστικών εισόδου και τέλος, η μόνη παράμετρος που χρειάζεται να προσδιορισθεί είναι η παράμετρος κόστους  $C$  από τον χρήστη. Από την άλλη πλευρά υπάρχουν και κάποια μειονεκτήματα, τα οποία όμως μπορούν να επιλυθούν με την κατάλληλη αντιμετώπιση.

Ένα από αυτά είναι η πως ο ρυθμός εκπαίδευσης μειώνεται καθώς το σύνολο των δεδομένων αυξάνεται. Για το λόγο αυτό χρησιμοποιούνται γρήγοροι αλγόριθμοι επίλυσης προβλημάτων τετραγωνικού προγραμματισμού αντί της αναλυτικής λύσης των SVM. Επιπλέον, όταν ο αριθμός των διανυσμάτων υποστήριξης είναι μεγάλος, η αξιολόγηση των άγνωστων δειγμάτων είναι επίσης αργή. Μία λύση είναι όπως αναφέρει ο Burges να απλοποιηθούν οι κανόνες απόφασης (Burges & Schölkopf, 1997). Τέλος, με την εφαρμογή τεχνικών όπως οι ένας-εναντίον-ενός, ένας-εναντίον-όλων καταπολεμάται το πρόβλημα που αφορά ότι τα SVM αντιμετωπίζουν προβλήματα μόνο δύο κλάσεων.

## 5.5 Αλγόριθμος Καταιγισμού Ιδεών και κατηγοριοποίηση

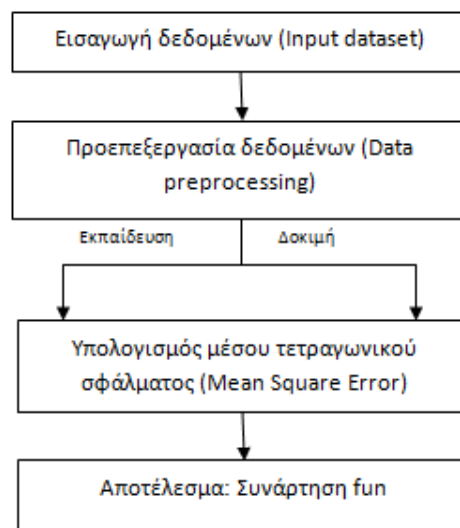
### δεδομένων

Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών μπορεί να επιφέρει πλεονεκτήματα σε μεγάλη ποικιλία εφαρμογών. Μία εφαρμογή που υλοποιήθηκε στην παρούσα εργασία είναι η εύρεση των βέλτιστων τιμών των παραμέτρων  $C, \gamma$  στα SVM προκειμένου να επιτευχθεί η βέλτιστη δυνατή κατηγοριοποίηση δεδομένων.

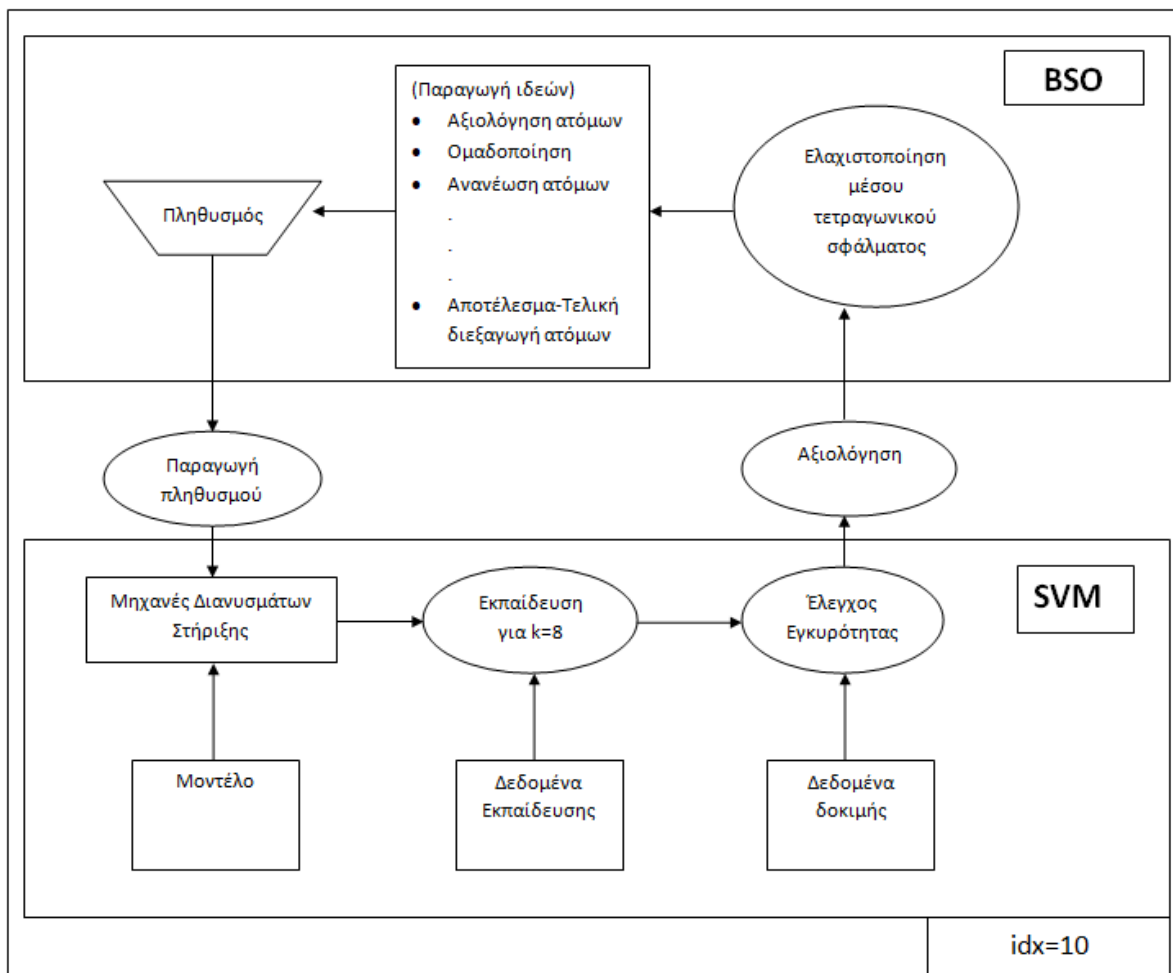
Αρχικά, εισάγονται δεδομένα τα οποία και προεπεξεργάζονται προκειμένου να χρησιμοποιηθούν. Αυτό σημαίνει πως κάποια από τα δεδομένα χωρίζονται με σκοπό την εκπαίδευση (*train*), ενώ άλλα αποσκοπούν στη δοκιμή (*test*). Ολόκληρη η διαδικασία επαναλαμβάνεται για 10 επαναλήψεις, προκειμένου να ελεγχθεί η εγκυρότητα και η αντιπροσωπευτικότητα των αποτελεσμάτων. Σε κάθε μία επανάληψη, διαφορετικά δεδομένα για δοκιμή και εκπαίδευση επιλέγονται τυχαία, χρησιμοποιώντας k-fold cross validation μέθοδο, για  $k=8$ . Στη συνέχεια

πραγματοποιείται ο υπολογισμός του μέσου τετραγωνικού σφάλματος (*Mean Square Error* (MSE)) που αποτελεί τη συνάρτηση  $f_{in}$ , του σχήματος της **Εικόνας 14**.

Αφού έχει πραγματοποιηθεί η διαδικασία παραγωγής της συνάρτησης MSE, ο σκοπός πλέον είναι η ελαχιστοποίησή της, έτσι ώστε να διεξαχθεί το βέλτιστο δυνατό αποτέλεσμα. Γι' αυτό το λόγο γίνεται στη συνέχεια η χρήση του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών, ο οποίος βρίσκει τις βέλτιστες τιμές των μεταβλητών  $C$  και  $\gamma$ , οι οποίες θα ελαχιστοποιήσουν, κατά το μέγιστο δυνατό, το μέσο τετραγωνικό σφάλμα. Ο αλγόριθμος, στην πραγματικότητα, αξιολογεί ζευγάρια  $C, \gamma$ , που αποτελούν τα άτομα του πληθυσμού, όπως αναφέρθηκε και σε παραπάνω ενότητα, τα ομαδοποιεί σε πέντε κατηγορίες, όσες και τα κέντρα που έχουν οριστεί και επιλέγει το μικρότερο μέσο τετραγωνικό σφάλμα. Στην **Εικόνα 14** παρουσιάζεται διαγραμματικά και συνοπτικά η διαδικασία διεξαγωγής της συνάρτησης μέσου τετραγωνικού σφάλματος, η οποία στη συνέχεια θα χρησιμοποιηθεί από τον BSO, όπως φαίνεται στην **Εικόνα 15**, και θα χρησιμοποιηθεί στα SVM, προκειμένου να βρεθεί η βέλτιστη τιμή των  $C, \gamma$  για κατηγοριοποίηση δεδομένων.



**Εικόνα 14** Διάγραμμα Ροής Συσχετίσεων για Μηχανές Διανυσμάτων Στήριξης



Εικόνα 15 Συνοπτικό διάγραμμα εφαρμογής 1 BSO-SVM

**Πίνακας 7 Περιγραφή δεδομένων από το UCI**

Δεδομένα Dataset	Αριθμός χαρακτηριστικών No. features	Αριθμός οντοτήτων No. instances	Αριθμός κλάσεων (No. classes)	Άγνωστες τιμές (Missing values)
Glass Identification	10	214	6	-

Ο καθορισμός των ορίων των παραμέτρων  $C, \gamma$  έγινε με αναζήτηση πλέγματος (*grid search*) (Hsu, Chang, Lin, & others, 2003). Έτσι, το  $C \in [2^{-5}, 2^{15}]$  ενώ το  $\gamma \in [2^{-15}, 2^3]$ . Έπειτα από αρχικοποίηση των παραμέτρων με τις τιμές που καταγράφονται στον **Πίνακα 4**, ο BSO δοκιμάστηκε για πληθυσμό μεγέθους  $N=20$ , που αποτελούν τα άτομα και αντιστοιχούν στις πιθανές τιμές των  $C, \gamma$ . Οι λύσεις αυτές ομαδοποιήθηκαν σε  $n_c=5$  συστάδες, μέσω του αλγορίθμου K-μέσου. Παρακάτω, φαίνονται τα αποτελέσματα του αλγορίθμου BSO με το SVM, ως προς τη μέση τιμή και την τυπική απόκλιση. Στον **Πίνακα 8**, παρουσιάζονται επίσης οι εφαρμογές κατηγοριοποίησης των ίδιων δεδομένων με διάφορους αλγορίθμους, όπως αναγράφονται στη δημοσίευση των (Sayed, Soliman, & Hassanien, 2018), ώστε να μπορέσουν να γίνουν κατάλληλες συγκρίσεις. Όσον αφορά τα δεδομένα που χρησιμοποιήθηκαν, αντλήθηκαν από το UCI και ονομάζονται "GlassDataIdentification". Περιέχουν 214 οντότητες, 10 χαρακτηριστικά και αποτελείται από 6 τύπους γυαλιού οι οποίοι καθορίζονται βάση το περιεχόμενο οξειδίου τους (Na, Mg, Al, Si, K, Ca, Ba, Fe).

Όπως φαίνεται στον **Πίνακα 8** στη σύγκριση συμμετέχουν ο Αλγόριθμος βελτιστοποίησης Αποικίας Μελισσών (Artificial Bee Colony (ABC)) με τη μέση τιμή να ισούται με 28.122 και τυπική απόκλιση 8.292, ο Γενετικός Αλγόριθμος (Genetic Algorithm (GA)) με μέση τιμή 33.347 και τυπική απόκλιση 15.798, ο Αλγόριθμος βελτιστοποίησης Πουλερικών (Chicken Swarm Optimization (CSO)) με μέση τιμή 26.204 και τυπική απόκλιση 3.892, ο Αλγόριθμος βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization (PSO)) με μέση τιμή 67.579 και τυπική απόκλιση 16.593, ο Αλγόριθμος βελτιστοποίησης Νυχτερίδας (Bat Algorithm (BA)) με μέση τιμή 34.616 και τυπική απόκλιση 10.965 και ο Αλγόριθμος βέλτιστης Αναζήτησης Τροφής (Optimal Foraging Algorithm (OFA)) με μέση τιμή 23.545 και τυπική απόκλιση 4.876.

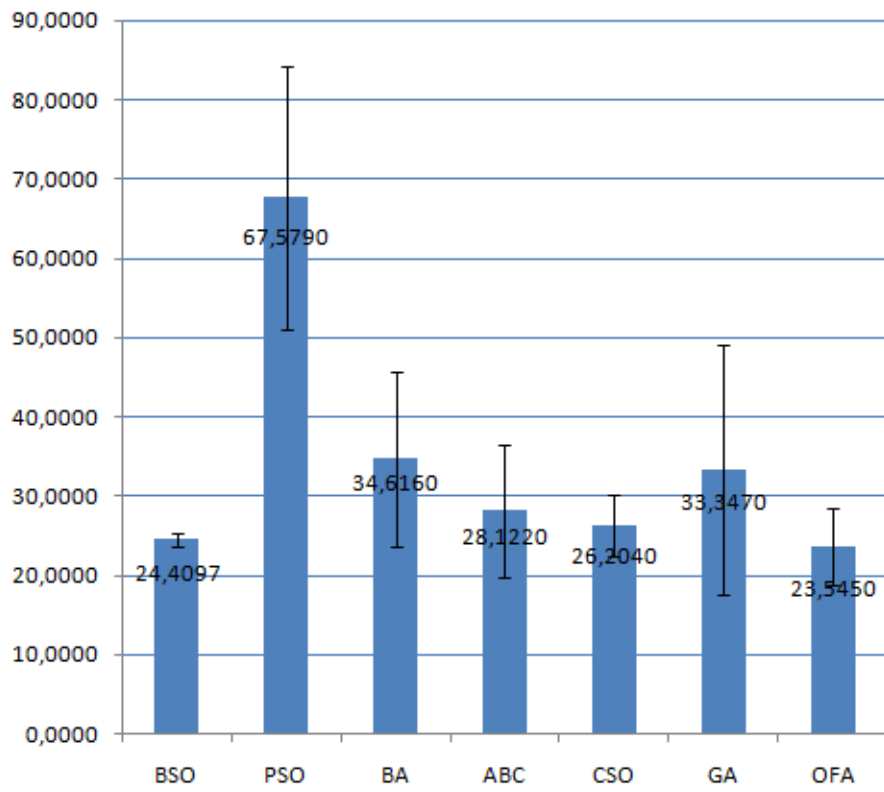
Συμπερασματικά, όπως είναι φανερό, συγκριτικά με τους άλλους αλγορίθμους, ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών, κατά βάση, επιφέρει καλύτερα αποτελέσματα, καθώς η βέλτιστη μέση τιμή μέσου τετραγωνικού σφάλματος ισούται με 24.4097 και η τυπική απόκλιση με 0.8145, για τις τιμές  $C=21.531$  και  $\gamma=5$ . Όσο μικρότερη είναι η μέση τιμή του μέσου τετραγωνικού σφάλματος και η τυπική απόκλιση, τόσο καλύτερα γίνεται η κατηγοριοποίηση δεδομένων και επομένως χαρακτηρίζεται ο αλγόριθμος αποδοτικός.

**Πίνακας 8** Αποτελέσματα υπολογισμού μέσης τιμής και τυπικής απόκλισης μέσου τετραγωνικού σφάλματος με τη βοήθεια βιο-εμπνευσμένων αλγορίθμων

Αλγόριθμος	Μέση τιμή	Τυπική απόκλιση
<b>BSO</b>	<b>24.4097</b>	<b>0.8145</b>
PSO	67.579	16.593
BA	34.616	10.965
ABC	28.122	8.292
CSO	26.204	3.892
GA	33.347	15.798
OFA	23.545	4.876

Στον **Πίνακα 9** απεικονίζεται ένα ραβδόγραμμα με το μέσο τετραγωνικό σφάλμα και γραμμές σφάλματος, που αντιπροσωπεύουν την τυπική απόκλιση. Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών, κατά βάση, επιφέρει καλύτερα αποτελέσματα, καθώς η μέση τιμή του ισούται με 24.4097 και η τυπική απόκλιση με 0.8145. Όσο μικρότερη είναι η μέση τιμή τετραγωνικού σφάλματος και η τυπική απόκλιση, τόσο καλύτερη γίνεται η κατηγοριοποίηση δεδομένων και επομένως χαρακτηρίζεται ο αλγόριθμος αποδοτικός. Επιπλέον, συγκρίνοντας την τυπική απόκλιση του BSO με τους υπόλοιπους αλγορίθμους, παρατηρείται ότι ο αλγόριθμος αυτός είναι σταθερός. Στον **Πίνακα 9** μελετώνται οι παρακάτω αλγόριθμοι: Αλγόριθμος Καταιγισμού Ιδεών (Brain Storm Optimization Algorithm (BSO)), Αλγόριθμος βελτιστοποίησης αποικίας μελισσών (Artificial Bee Colony (ABC)), Γενετικός Αλγόριθμος (Genetic Algorithm (GA)), Αλγόριθμος βελτιστοποίησης πουλερικών (Chicken Swarm Optimization (CSO)), Αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization (PSO)), Αλγόριθμος βελτιστοποίησης νυχτερίδας (Bat Algorithm (BA)), Αλγόριθμος βέλτιστης αναζήτησης τροφής (Optimal Foraging Algorithm (OFA))

Πίνακας 9 Διάγραμμα μέσης τιμής μέσου τετραγωνικού σφάλματος συσχετισμένο με τη τυπική απόκλιση





# 6.Βελτιστοποίηση Καταιγισμού Ιδεών για τη Μείωση του Θορύβου σε Εικόνες

Η εξέλιξη της τεχνολογίας και ειδικότερα των υπολογιστών έχει επιφέρει μεγάλη βελτίωση στην επεξεργασία των εικόνων. Η ψηφιακή επεξεργασία εικόνας αποτελεί πλέον ολόκληρη επιστήμη και έχει ευρύτατες εφαρμογές σε άλλους επιστημονικούς κλάδους όπως για παράδειγμα στη ρομποτική και στην όραση μηχανής (*computer vision*).

Κατά τη διαδικασία δημιουργίας, αποθήκευσης, μετάδοσης και επεξεργασίας μίας ψηφιακής εικόνας υπεισέρχονται ανεπιθύμητες αλλοιώσεις αυτής, που έχουν ως αποτέλεσμα την υποβάθμιση της ποιότητάς της. Οι αλλοιώσεις αυτές μπορεί να οφείλονται σε διάφορα είδη θορύβου (Παπαμαρκος, 2015).

## 6.2 Θόρυβος στις εικόνες

Υπάρχουν διάφορα είδη θορύβου που συναντώνται σε μία ψηφιακή εικόνα και ποικίλουν ανάλογα με τον τρόπο δημιουργίας της εκάστοτε εικόνας(Παπαμαρκος, 2015). Υπάρχει πιθανότητα μία εικόνα να έχει προκύψει, για παράδειγμα, από τη σάρωση ενός film. Τότε, ο κόκκος του film είναι πηγή θορύβου. Η επίπτωση του θορύβου στις εικόνες μπορεί να περιγραφεί από το λόγο του σήματος προς θόρυβο (*SNR- signal to noise ratio*). Το SNR μπορεί να οριστεί και ως εξής:

$$SNR = \frac{\sigma_s}{\sigma_n} = \sqrt{\frac{\sigma_f^2}{\sigma_n^2} - 1} \quad (6.1)$$

όπου  $\sigma_s^2$  και  $\sigma_f^2$  είναι οι μεταβλητότητες της αρχικής εικόνας και της εικόνας με το θόρυβο, αντίστοιχα. Η μεταβλητή  $\sigma_n$  χαρακτηρίζει τη μεταβλητότητα του προσθετικού θορύβου, ο οποίος συνήθως έχει μηδενική μέση τιμή.

Επιπρόσθετα, ένας άλλος δείκτης είναι το PSNR (*peak to noise ratio*), ο οποίος υπολογίζει σε decibels το λόγο του μέγιστου σήματος προς το θόρυβο μεταξύ δύο εικόνων. Σημαντικό είναι να αναφερθεί ότι όσο υψηλότερο είναι το PSNR, τόσο λιγότερο θόρυβος υπάρχει στην εικόνα. Το PSNR ορίζεται ως εξής:

$$PSNR = 10 \log \frac{D^2}{MSE} \quad (6.2)$$

Στην παραπάνω εξίσωση, το D είναι ίσο με τη μέγιστη διακύμανση της αρχικής εικόνας. Όταν η εικόνα είναι σε αποχρώσεις του γκρι, η τιμή κάθε εικονοστοιχείου αποτυπώνεται με 8 bits και η τιμή του D είναι  $D = 2^8 - 1 = 255$ . Στις έγχρωμες εικόνες, το D καθορίζεται με τον ίδιο τρόπο όμως το MSE ισούται με τη μέση τιμή στις τρεις συνιστώσες.

Έτσι, προκύπτει ο τύπος:

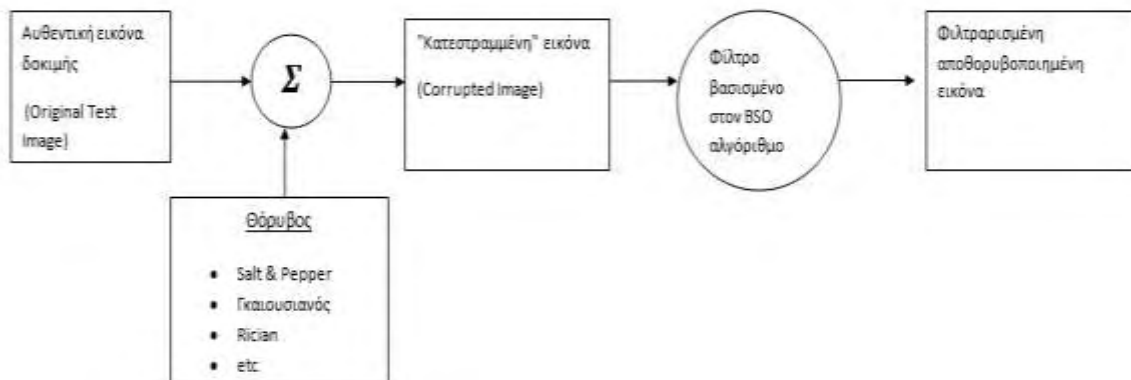
$$PSNR = 10 \log \frac{D^2}{\frac{MSE(R)+MSE(G)+MSE(B)}{3}} \quad (6.3)$$

Επιπρόσθετα, το MSE βρίσκεται από τον παρακάτω τύπο:

$$MSE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N [I(m, n) - I'(m, n)]^2 \quad (6.4)$$

Όπου  $I(m,n)$  η φωτεινότητα του pixel  $(m,n)$  της αρχικής εικόνας και  $I'$  η φωτεινότητα του ίδιου pixel της θορυβοποιημένης εικόνας.

## 6.3 Αλγόριθμος καταίγισμού ιδεών και αποθορυβοποίηση εικόνων



Εικόνα 16 Διάγραμμα αποθορυβοποίησης εικόνας με τον BSO

### 6.3.1 Αποθορυβοποίηση εικόνων με φίλτρο ενδιάμεσης τιμής

Υπάρχουν διάφορα είδη φίλτρων με τα οποία μπορεί να επιτευχθεί η αποθορυβοποίηση μίας εικόνας (Παπαμαρκος, 2015). Κάποια χαρακτηριστικά παραδείγματα φίλτρων είναι τα παρακάτω:

- Φίλτρα μέσης τιμής (*mean filter*)
- Το φίλτρα ενδιάμεσης τιμής (*median filter*)
- Το φίλτρα εξομάλυνσης με συντήρηση των ακμών (*edge preserving filters*)
- Φίλτρα ελαχίστου/μεγίστου (*min/max filters*)
- Πολυκλιμακωτά φίλτρα ενδιάμεσης τιμής (*max/median filters*)
- Φίλτρα Gauss

Στην παρούσα εφαρμογή έγινε χρήση του φίλτρου ενδιάμεσης τιμής, γι' αυτό και θα δοθεί παραπάνω έμφαση σε αυτό.

Η ενδιάμεση τιμή συνόλου  $A$  είναι ίση με τη ενδιάμεση τιμή του συνόλου. Σε ένα σύνολο το οποίο έχει περιττό αριθμό στοιχείων, ταξινομούνται τα στοιχεία του συνόλου κατά αύξουσα τιμή,

οπότε η ενδιάμεση τιμή του συνόλου ισούται με το μεσαίο στοιχείο αν έχουμε περιττό αριθμό στοιχείων ή με τη μέση τιμή των δύο γειτονικών μεσαίων στοιχείων στην περίπτωση άρτιου αριθμού.

Για παράδειγμα, έστω

$$A = \{a_1, a_2, \dots, a_n\}$$

το σύνολο με στοιχεία  $a_1 \leq a_2 \leq \dots \leq a_n \in \mathbb{R}$ . Η ενδιάμεση τιμή του  $A$  ισούται με:

$$\text{median}(A) = \begin{cases} \frac{a_{n+1}}{2}, & \text{αν } n \text{ περιττός} \\ \frac{1}{2} \left( \frac{a_n}{2} + \frac{a_{n+1}}{2} \right), & \text{αν } n \text{ άρτιος} \end{cases} \quad (6.5)$$

Από τον ορισμό της ενδιάμεσης τιμής προκύπτει ότι αν  $k$  σταθερά τότε

$$\text{median}(k+A) = k + \text{median}(A) \quad (6.6)$$

$$\text{median}(k \times A) = k \times \text{median}(A) \quad (6.7)$$

Στη Matlab, όπου υλοποιήθηκε το πρόγραμμα, η εντολή που δίνει το φίλτρο ενδιάμεσης τιμής είναι:  $B = \text{medfilt2}(A, [m \ n])$  και εκτελεί φίλτρο με γειτονιά  $m \times n$ . Δηλαδή, η τιμή κάθε εικονοστοιχείου  $B(x,y)$ , που προκύπτει, ισούται με τη μεσαία τιμή της γειτονιάς  $m \times n$  γύρω από το αντίστοιχο εικονοστοιχείο  $A(x,y)$ . Αν δεν υπάρχει συγκεκριμένος καθορισμός για τη γειτονιά, τότε ως default επιλογή διαστάσεων είναι η  $m \times n = 3 \times 3$ .

Στην παρούσα εφαρμογή, υλοποιείται η αποθορυβοποίηση εικόνας με τη χρήση φίλτρου ενδιάμεσης τιμής. Αρχικά, γίνεται αρχικοποίηση των μεταβλητών όπως φαίνεται στον **Πίνακα 4**. Στη συνέχεια ξεκινάει η επεξεργασία της εικόνας, η οποία περιέχει ήδη θόρυβο. Ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών εφαρμόζεται προκειμένου να βρει τις βέλτιστες τιμές της γειτονιάς  $m \times n$  του φίλτρου ενδιάμεσης τιμής, οι οποίες θα πετύχουν το ελάχιστο μέσο τετραγωνικό σφάλμα. Δεδομένου ότι τα ποσά MSE και PSNR είναι αντιστρόφως ανάλογα, όσο μικρότερο είναι το MSE, τόσο μεγιστοποιείται το PSNR και επομένως προκύπτει καλύτερη αποθορυβοποίηση της εικόνας. (6.2).

Ο αλγόριθμος BSO δοκιμάστηκε σε τρεις διαφορετικές εικόνες, προκειμένου να αξιολογηθεί η αποδοτικότητά του. Αρχικά, στην εικόνα της Lena, η βέλτιστη τιμή του PSNR βρέθηκε ίση με 31.4239, για φίλτρο ενδιάμεσης τιμής διάστασης  $5 \times 3$  και επίπεδο θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος καταγράφονται στον **Πίνακα 10**, ενώ η **Εικόνα 18(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 17**. Στη συνέχεια, δοκιμάστηκε η εικόνα του Cameraman. Η βέλτιστη τιμή του PSNR βρέθηκε ίση με 24.8349, για φίλτρο ενδιάμεσης τιμής διάστασης  $4 \times 3$  και επίπεδο θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος βρίσκονται στον **Πίνακα 11**, ενώ η **Εικόνα 20(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 19**. Τέλος, στην εικόνα Peppers, η βέλτιστη τιμή του PSNR βρέθηκε ίση με 34.4978, για φίλτρο ενδιάμεσης τιμής διάστασης  $5 \times 3$  και επίπεδο

θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος καταγράφονται στον **Πίνακα 12**, ενώ η **Εικόνα 22(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 21**.

Βλέποντας τα αποτελέσματα, παρατηρείται και στις τρεις περιπτώσεις η μέγιστη τιμή του PSNR στο επίπεδο θορύβου 0.005. Μάλιστα, η μέγιστη τιμή εκ των τριών πειραμάτων είναι στην τρίτη περίπτωση, ενώ συμπεραίνεται πως η τιμή αυτή μεταβάλλεται ανάλογα με το περιεχόμενο της κάθε εικόνας.



**Εικόνα 17 Αρχική εικόνα**



**Εικόνα 18(α)** Θορυβοποιημένη εικόνα (αριστερά) | **(β)** Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής 5×3 και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

**Πίνακας 10** Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής N×M στην εικόνα της Lena

Noise Variation	PSNR	N	M
0.005	<b>31.4239</b>	<b>5</b>	<b>3</b>
0.01	25.4917	9	9
0.015	28.4665	7	5
0.02	27.5590	11	5

0.025	26.6646	3	8
0.03	27.1241	7	4
0.035	26.3126	3	4
0.04	26.8159	9	7
0.045	26.3567	3	4
0.05	25.8179	8	9



Εικόνα 19 Αρχική εικόνα



Εικόνα 20 (α) Θορυβοποιημένη εικόνα (αριστερά) | (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής 3×3 και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

Πίνακας 11 Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής N×M στην εικόνα Cameraman.

Noise Variation	PSNR	N	M
0.005	<b>26.3518</b>	<b>3</b>	<b>3</b>
0.01	24.2884	3	5
0.015	23.1522	5	5
0.02	23.4900	7	1

0.025	23.0977	3	5
0.03	23.5789	3	3
0.035	22.8155	7	3
0.04	22.1774	8	9
0.045	21.0512	5	8
0.05	21.5400	3	7



Εικόνα 21 Αρχική εικόνα



Εικόνα 22(α) Θορυβοποιημένη εικόνα (αριστερά) | (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με χρήση φίλτρου ενδιάμεσης τιμής 5×3 και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

Πίνακας 12 Αποτελέσματα PSNR, έπειτα από εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών για αποθορυβοποίηση εικόνας με χρήση φίλτρου ενδιάμεσης τιμής N×M στην εικόνα Peppers.

Noise Variation	PSNR	N	M
0.005	34.4978	5	3
0.01	33.1604	5	3
0.015	32.0968	5	3
0.02	31.2704	5	3
0.025	30.1315	4	7
0.03	30.0801	5	9
0.035	29.1168	9	6
0.04	29.8388	7	7
0.045	29.3056	7	9
0.05	29.2072	7	6

### 6.3.2 Αποθορυβοποίηση εικόνων με χρήση μετασχηματισμού κυματιδίου

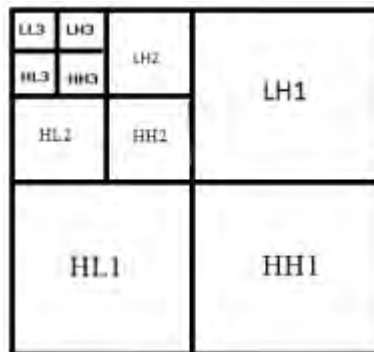
Ο διδιάστατος διακριτός μετασχηματισμός κυματιδίου (2D Discrete Wavelet Transform- 2D-WT) είναι διαχωρίσιμος και μπορεί να εφαρμοστεί ξεχωριστά σε γραμμές και στήλες. (Εικόνα 23) Στις δύο διαστάσεις, απαιτείται η χρήση μίας διδιάστατης συνάρτησης κλιμάκωσης  $\varphi(x, y)$  και τριών διδιάστατων συναρτήσεων κυματιδίων  $\psi^H(x, y), \psi^V(x, y), \psi^D(x, y)$  (Rafael Gonzalez, 2008). Σε κάθε ένα στάδιο του μετασχηματισμού κυματιδίου, η εικόνα αναλύεται σε τέσσερις εικόνες έχοντας ως μέγεθος το ένα τέταρτο του μεγέθους της αρχικής εικόνας (Εικόνα 22).

$$\psi^H(x, y) = \psi(x)\varphi(y) \tag{6.8}$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \tag{6.9}$$

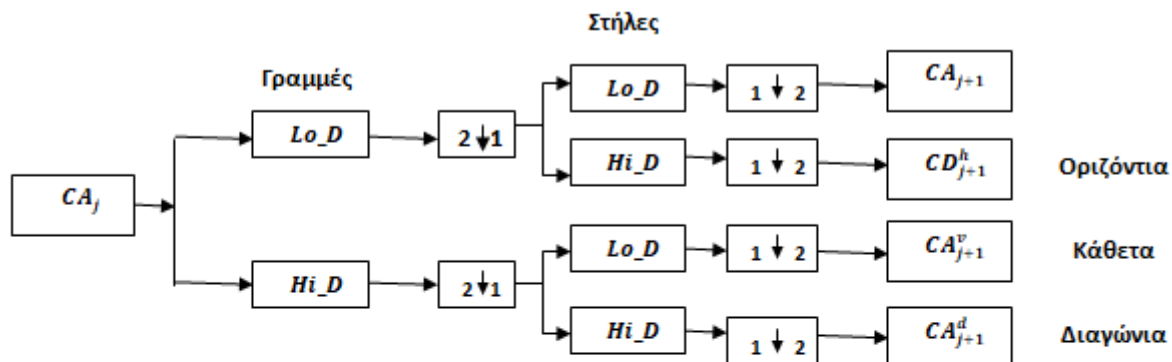
$$\psi^D(x, y) = \psi(x)\psi(y) \tag{6.10}$$

Οι σχέσεις (6.7)-(6.9) επιτρέπουν την καταμέτρηση των συναρτησιακών μεταβολών. Πιο συγκεκριμένα κατά μήκος διαφορετικών κατευθύνσεων: το κυματίδιο  $\psi^H$  μετράει τις μεταβολές κατά μήκος των στηλών, το κυματίδιο  $\psi^V$  μετράει τις μεταβολές κατά μήκος των γραμμών και το κυματίδιο  $\psi^D$  μετράει τις μεταβολές κατά μήκος διαγώνιων διευθύνσεων. Η κατευθυντική αυτή ευαισθησία, ανακλύπει ως μία φυσική συνέπεια της διαχωρισιμότητας του μετασχηματισμού και δεν αυξάνει την υπολογιστική του πολυπλοκότητα. Επιπρόσθετα, η διδιάστατη έκδοση του διακριτού μετασχηματισμού κυματιδίου μπορεί να υλοποιηθεί χρησιμοποιώντας ψηφιακά φίλτρα και συστήματα υποδειγματοληψίας, χρησιμοποιώντας διαχωρίσιμες διδιάστατες συναρτήσεις κλιμάκωσης και κυματιδίων.



Εικόνα 23Ο διδιάστατος διακριτός μετασχηματισμός κυματιδίου τριών επιπέδων





όπου

- $2\downarrow 1$  Υποδειγματοληψία στηλών, διατηρώντας τις άρτιες στήλες
- $1\downarrow 2$  Υποδειγματοληψία γραμμών, διατηρώντας τις άρτιες γραμμές
- $X$  (γραμμές): Συνέλιξη γραμμών εισόδου με φίλτρο  $X$
- $X$  (στήλες): Συνέλιξη στηλών εισόδου με φίλτρο  $X$

Αρχικοποίηση  $CA_0 = s$  για αρχικοποίηση αποσύνθεσης

Εικόνα 24 Βασικά βήματα Διδιάστατου Διακριτού Μετασχηματισμού Κυματιδίου αποσύνθεσης εικόνας

Η αποθορυβοποίηση της εικόνας, σε αυτή την εφαρμογή, γίνεται με χρήση του κυματιδίου Haar (Παπαμαρκος, 2015). Εκτός από την απομάκρυνση θορύβου σε δεδομένα, ο μετασχηματισμός Haar βρίσκει εφαρμογή στη συμπίεση εικόνων, βίντεο καθώς και στην εξαγωγή χαρακτηριστικών υφής. Αποτελεί την πρώτη θεώρηση κυματιδίων (*wavelets*) και είναι το απλούστερο δυνατό κυματίδιο. Το κύριο πλεονέκτημα του μετασχηματισμού αυτού είναι ότι δειγματοληπτεί τα δεδομένα εισόδου σε πολλαπλές αναλύσεις, με χαμηλή έως υψηλή ανάλυση και υπολογίζει τις διαφορές γειτονικών ζευγών.

Επιπρόσθετα, το υπολογιστικό κόστος του μετασχηματισμού είναι σχετικά μικρό αφού ο μετασχηματισμός ενός διανύσματος  $N$  στοιχείων απαιτεί μόνο  $2(N-1)$  προσθέσεις και  $N$  πολλαπλασιασμούς.

Ο μετασχηματισμός Haar βασίζεται στις συναρτήσεις Haar  $\square_k(x)$  οι οποίες ορίζονται στο διάστημα  $[0,1]$  για  $k = 0, 1, \dots, N-1$  όπου  $N = 2^n$  με  $n=1, 2, 3, \dots$ . Ισοδύναμα το  $k$  γράφεται

$$k = 2^p + q - 1 \quad (6.11)$$

Επομένως, συγκεντρωτικά οι συναρτήσεις του Haar είναι οι παρακάτω:

$$h_0(x) = h_{00}(x) = \frac{1}{\sqrt{N}}, 0 \leq x \leq 1 \quad (6.12)$$

$$h_k(x) = h_{pq}(x) = \frac{1}{\sqrt{N}} \begin{cases} (\sqrt{2})^p, & \frac{(q-1)}{2^p} \leq x \leq \frac{q-1}{2^p} \\ -(\sqrt{2})^p, & \frac{(q-1)}{2^p} \leq x \leq \frac{q}{2^p} \\ 0, & \text{διαφορετικά για } x \in [0,1] \end{cases} \quad (6.13)$$

Προκειμένου να προσδιοριστούν οι συντελεστές του Διδιάστατου Διακριτού Μετασχηματισμού Κυματιδίου, θα πρέπει να γίνει χρήση ενός κατωφλίου (threshold). Στη συγκεκριμένη εφαρμογή χρησιμοποιούνται δύο είδη κατωφλίσιας: το soft και το hard, οι τύποι των οποίων δίνονται παρακάτω.

Έτσι, για το soft threshold, ο τύπος είναι:

$$x' = \begin{cases} x, & |x| \geq T \\ 0, & |x| < T \end{cases} \quad (6.14)$$

ενώ για το hard threshold, ο τύπος είναι:

$$x' = \begin{cases} \text{sgn}(x)(|x| - T), & |x| \geq T \\ 0, & |x| < T \end{cases} \quad (6.15)$$

όπου  $x$  είναι το σήμα και  $T$  η τιμή του κατωφλίου.

Στην παρούσα εφαρμογή, υλοποιείται η αποθορυβοποίηση εικόνας με τη χρήση διακριτού μετασχηματισμού σωματιδίου στις δύο διαστάσεις. Ο αλγόριθμος εφαρμόζεται σε τρεις περιπτώσεις, στις εικόνες Lena, Cameraman και Peppers. Αρχικά, γίνεται αρχικοποίηση των μεταβλητών όπως φαίνεται στον **Πίνακα 4**, ενώ επιλέγεται το κυματίδιο Haar, ώστε να σχηματιστεί η αλλοιωμένη με θόρυβο εικόνα. Στη συνέχεια εφαρμόζεται στην εικόνα κατώφλι τύπου soft και hard. Θα πρέπει να επισημανθεί πως η εφαρμογή κατωφλίου έγινε μόνο στους συντελεστές λεπτομέρειας (*detailed coefficients*) του διακριτού διδιάστατου μετασχηματισμού κυματιδίου, όπως επίσης εφαρμόστηκε και στη δημοσίευση των (Sivaranjani, Roomi, & Senthilarasi, 2019), προκειμένου να μπορέσει να διεξαχθεί σύγκριση των αποτελεσμάτων. Ειδικότερα, ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών βρίσκει τις βέλτιστες τιμές των συντελεστών λεπτομέρειας, οι οποίες θα είναι τέτοιες ώστε να επιτύχουν το ελάχιστο MSE και το μέγιστο PSNR, αφού όπως αναφέρθηκε και στον τύπο (6.2), τα ποσά αυτά είναι αντιστρόφως ανάλογα. Τέλος, πραγματοποιείται ο υπολογισμός του αντίστροφου διακριτού μετασχηματισμού κυματιδίου, δηλαδή πραγματοποιείται μία ανακατασκευή των κυματιδίων.

Αρχικά, στην εικόνα της Lena, η βέλτιστη τιμή του PSNR βρέθηκε ίση με 31.5976, για φίλτρο soft και επίπεδο θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος καταγράφονται στον **Πίνακα 13**, ενώ η **Εικόνα 25(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 24**. Στη συνέχεια, δοκιμάστηκε η εικόνα του Cameraman. Η βέλτιστη τιμή του PSNR βρέθηκε ίση με 28.1587, για φίλτρο hard και επίπεδο θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος βρίσκονται στον **Πίνακα 15**, ενώ η **Εικόνα 27(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 26**. Τέλος, στην εικόνα Peppers, η βέλτιστη τιμή του PSNR βρέθηκε ίση με 34.0928, για φίλτρο soft και επίπεδο θορύβου 0.005. Τα αποτελέσματα του αντίστοιχου πειράματος καταγράφονται στον **Πίνακα 16**, ενώ η **Εικόνα 29(β)** αποτελεί τη βέλτιστη και αποθορυβοποιημένη εκδοχή της αρχικής εικόνας, **Εικόνα 28**.

Βλέποντας τα αποτελέσματα, παρατηρείται και στις τρεις περιπτώσεις η μέγιστη τιμή του PSNR να είναι στο επίπεδο θορύβου 0.005. Μάλιστα, η μέγιστη τιμή εκ των τριών πειραμάτων είναι στην τρίτη περίπτωση, ενώ συμπεραίνεται πως η τιμή αυτή μεταβάλλεται ανάλογα με το περιεχόμενο της κάθε εικόνας.

Στην περίπτωση της Lena πραγματοποιείται μία επιπλέον σύγκριση με τη δημοσίευση των (Sivaranjani et al., 2019). Στην εργασία αυτή, στόχος είναι η αφαίρεση του κοκκώδους θορύβου Speckle, σε εικόνες τύπου SAR, με τη χρήση του Αλγορίθμου Βελτιστοποίησης Σμήνους (Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)). Πιο συγκεκριμένα, όπως χαρακτηριστικά αναφέρεται σκοπός είναι η εύρεση του κατάλληλου κατωφλίου, άμεσα, έπειτα από χρήση του αλγορίθμου που αναφέρθηκε, καθώς και δυαδικού δέντρου σύνθετου μετασχηματισμού κυματιδίου (Dual-Tree Complex Wavelet Transform (DTCWT)), στο πεδίο των συχνοτήτων **Πίνακας 14**. Προκειμένου να αξιολογηθούν τα αποτελέσματα γίνεται χρήση των μετρήσεων αναφοράς Peak Signal-to-Noise Ratio (PSNR) και του Mean Structural Similarity Index Metric (MSSIM). Συγκρίνοντας τα αποτελέσματα της προαναφερθείσας δημοσίευσης με την εφαρμογή της παρούσας πτυχιακής εργασίας, διεξάγεται το συμπέρασμα ότι ο αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών με τη χρήση διδιάστατου διακριτού μετασχηματισμού κυματιδίου, στην εικόνα της Lena, είναι πιο αποτελεσματικός, καθώς παράγεται μεγαλύτερο PSNR με τιμή **31.5976** στην εφαρμογή soft threshold, ενώ εφαρμόζοντας τον Αλγόριθμο Βελτιστοποίησης Σμήνους και soft threshold, η τιμή που προκύπτει είναι **28.009**.



**Εικόνα 25 Αρχική εικόνα**



Εικόνα 26 (α) Θορυβοποιημένη εικόνα (αριστερά) | (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

Πίνακας 13 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με hard threshold και soft threshold και διαφορετικά επίπεδα θορύβου, της εικόνας Lena.

Noise Variation	PSNR με Hard Threshold	PSNR με Soft Threshold
0.005	30.4881	<b>31.5976</b>
0.01	28.9398	29.0696
0.015	27.8076	28.0185
0.02	26.9591	27.1355
0.025	26.6639	26.9393
0.03	25.6476	25.7787
0.035	25.1896	25.2037
0.04	24.7643	24.8016
0.045	24.3811	24.4201
0.05	24.0321	24.0407

Πίνακας 14 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Multi-ObjectivePSO, με hard threshold και soft threshold, της εικόνας Lena. (Sivaranjanietal, 2019)

Είδος κατωφλίου	PSNR με Hard Threshold	PSNR με Soft Threshold
PSNR	28.7617	28.0098



Εικόνα 27 Αρχική εικόνα



Εικόνα 28 (α) Θορυβοποιημένη εικόνα (αριστερά) | (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

Πίνακας 15 Αποτελέσματα PSNR για αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με *hardthreshold* και *softthreshold* και διαφορετικά επίπεδα θορύβου, της εικόνας *cameran*.

Noise Variation	PSNR με Hard Threshold	PSNR με Soft Threshold
0.005	<b>28.1587</b>	27.8343
0.01	27.5951	28.0222
0.015	26.3614	25.9221
0.02	25.1000	25.2999
0.025	24.8405	25.1599
0.03	24.0610	23.9679
0.035	23.7606	23.7687
0.04	23.1205	23.3724
0.045	22.8832	22.8994
0.05	22.4605	22.6727



Εικόνα 29 Αρχική εικόνα



Εικόνα 30 (α) Θορυβοποιημένη εικόνα (αριστερά) | (β) Εφαρμογή Αλγορίθμου Βελτιστοποίησης Καταγισμού Ιδεών με Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου και παραγωγή αποθορυβοποιημένης εικόνας (δεξιά)

Πίνακας 16 Αποτελέσματα PSNRγια αποθορυβοποίηση εικόνας χρησιμοποιώντας τον αλγόριθμο Βελτιστοποίησης Καταγισμού Ιδεών και το Διδιάστατο Διακριτό Μετασχηματισμό Κυματιδίου, με *hardthreshold* και *softthreshold* και διαφορετικά επίπεδα θορύβου, της εικόνας *Cameraman*

Noise Variation	PSNR με Hard Threshold	PSNR με Soft Threshold
0.005	34.0856	<b>34.0928</b>
0.01	31.6857	31.9852
0.015	30.7386	30.9633
0.02	29.9235	30.1386
0.025	29.2739	29.5734
0.03	28.6674	28.9677
0.035	28.2441	28.4482
0.04	27.7674	27.9952
0.045	27.3775	27.9975
0.05	26.6399	27.0343

Παρακάτω, στον **Πίνακα 17**, παρουσιάζονται συγκεντρωτικά οι βέλτιστες τιμές των PSNR που βρέθηκαν στις παραπάνω εφαρμογές, όσον αφορά την αποθορυβοποίηση εικόνας.

**Πίνακας 17 Βέλτιστες τιμές PSNR, συγκεντρωτικά, για αποθορυβοποίηση εικόνων με εφαρμογή του BSO Αλγορίθμου**

<b>Εικόνα/Τεχνική</b>	<b>Φίλτρο ενδιάμεσης τιμής</b>	<b>Μετασχηματισμός κωματιδίου</b>
	<b>PSNR</b>	
Lena	31.4239	31.5976
Cameraman	26.3518	28.1587
Peppers	34.4978	34.0928

## 7. Συμπεράσματα

Η παρούσα πτυχιακή εργασία πραγματεύεται τη μελέτη των βιο-εμπνευσμένων αλγορίθμων και την εφαρμογή αυτών. Ειδικότερα, έπειτα από βιβλιογραφική αναζήτηση, η μελέτη επικεντρώνεται στον Αλγόριθμο Βελτιστοποίησης Καταιγισμού Ιδεών (Brain Storm Optimization Algorithm), ο οποίος είναι σχετικά πρόσφατος και όπως αποδείχθηκε, πολύ πλεονεκτικός και ωφέλιμος.

Όπως γίνεται αντιληπτό, ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών βρίσκει ήδη πολλές εφαρμογές σε πολλούς τομείς, όπως σε ενεργειακά προβλήματα (Ramanand et al., 2012) (Jadhav et al., 2012) (Arsuaga-Rios & Vega-Rodriguez, 2014), σε μηχανικά προβλήματα (Duan et al., 2013), στο σχεδιασμό δορυφόρων και πτήσεων (Duan et al., 2018; Qiu & Duan, 2014; Soyinka & Duan, 2016; C. Sun et al., 2013), σε οικονομικής φύσεως ζητήματα (Yanqiu Sun, 2014; Jianzhou Wang et al., 2016) καθώς και σε πολλαπλών μοντέλων βελτιστοποιήσεις (X. Guo et al., 2015). Αυτό έχει ως αποτέλεσμα, να γίνει κατανοητό πως ο παρών αλγόριθμος έχει απήχηση σε πολλές εφαρμογές, καθώς προτιμάται να εφαρμόζεται όλο και περισσότερο.

Πιο συγκεκριμένα, έπειτα από πειραματικές δοκιμές, ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών προτείνεται, αφού πραγματοποιεί επιτυχώς κατηγοριοποίηση δεδομένων με τη βοήθεια Μηχανών Διανυσμάτων Στήριξης, καθώς επίσης και αποθορυβοποίηση εικόνων, με αποτέλεσμα την παραγωγή καλύτερης ποιότητας νέας εικόνας με επιτυχή απομάκρυνση ανεπιθύμητων στοιχείων.

Τα αποτελέσματα και συμπεράσματα που λαμβάνονται, έπειτα από εφαρμογές του BSO, είναι τα εξής:

- Κατηγοριοποίηση δεδομένων με τη βοήθεια των Μηχανών Διανυσμάτων Στήριξης, έπειτα από εύρεση των κατάλληλων τιμών των παραμέτρων  $C$ ,  $\gamma$ , οι οποίες επιφέρουν το ελάχιστο μέσο τετραγωνικό σφάλμα
- Αποθορυβοποίηση εικόνων με χρήση φίλτρου ενδιάμεσης τιμής και εύρεση βέλτιστου PSNR, έπειτα από καθορισμό των διαστάσεων του φίλτρου
- Ο Αλγόριθμος Καταιγισμού Ιδεών βρέθηκε να είναι πιο αποτελεσματικός στην κατηγοριοποίηση δεδομένων για τις Μηχανές Διανυσμάτων Στήριξης, συγκριτικά με άλλους βιο-εμπνευσμένους αλγορίθμους, όπως, ο Αλγόριθμος Βελτιστοποίησης Αποικίας Μελισσών, ο Γενετικός Αλγόριθμος, ο Αλγόριθμος Βελτιστοποίησης Πουλερικών, ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων και ο Αλγόριθμος Βελτιστοποίησης Νυχτερίδας
- Ο Αλγόριθμος Καταιγισμού Ιδεών παρουσιάζει μεγάλη σταθερότητα, καθώς έχει μικρή τιμή τυπικής απόκλισης, στην εφαρμογή κατηγοριοποίησης δεδομένων για τις Μηχανές Διανυσμάτων Στήριξης
- Αποθορυβοποίηση εικόνων με χρήση Διδιάστατου Διακριτού Μετασχηματισμού Κυματιδίου και εύρεση βέλτιστου PSNR, έπειτα από καθορισμό κατάλληλης τιμής κατωφλίου
- Ο Αλγόριθμος Καταιγισμού Ιδεών πραγματοποιεί καλύτερη αποθορυβοποίηση εικόνων συγκριτικά με τον Αλγόριθμο Βελτιστοποίησης Σμήνους



Συνοψίζοντας, κρίνεται, λοιπόν, ωφέλιμο να υπογραμμιστεί, ότι οι περισσότερες εφαρμογές δείχνουν αρκετά ελπιδοφόρα αποτελέσματα, στο πλαίσιο της συγκεκριμένης μελέτης, καθώς ο Αλγόριθμος Βελτιστοποίησης Καταιγισμού Ιδεών είναι ιδιαίτερα αποδοτικός, συγκριτικά με άλλους αλγορίθμους, τόσο για την εύρεση βέλτιστων παραμέτρων στις Μηχανές Διανυσμάτων Στήριξης για κατηγοριοποίηση δεδομένων, όσο και για την αποθορυβοποίηση εικόνων.

Ανάμεσα στα μελλοντικά πλάνα της έρευνας στο πλαίσιο αυτό, είναι η εφαρμογή του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών για αποθορυβοποίηση ηχητικών σημάτων ιατρικών δεδομένων, όπως σήματα ηλεκτροκαρδιογραφήματος. Αυτό θα οδηγήσει στην απομάκρυνση ανεπιθύμητου θορύβου διαφόρων ηχητικών κυμάτων, με αποτέλεσμα να μπορέσει να εξεταστεί και να διαγνωστεί καλύτερα κάποιο πιθανό πρόβλημα. Επιπρόσθετα, μία άλλη εφαρμογή θα είναι ο υβριδισμός του Αλγορίθμου Βελτιστοποίησης Καταιγισμού Ιδεών με άλλους βιο-εμπνευσμένους αλγορίθμους, που δεν έχει ακόμα συνδυαστεί και η εφαρμογή του για κατηγοριοποίηση δεδομένων ή βελτιστοποίηση παραμέτρων συστήματος.

# Βιβλιογραφία

- AbWahab, M., Nefti-Meziani, S., & Atyabi, A. (2015).  
A comprehensive review of swarm optimization algorithms. *PLoS One*, *10*(5), e0122827.  
Public Library of Science.
- Afonso, L. C., Passos, L., & Paulo Papa, J. (2018). Enhancing Brain Storm Optimization Through Optimum-Path Forest. *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (pp. 183–188). IEEE.
- Aizerman, M. (1964).  
Theoretical foundations of the potential function method in pattern recognition learning.  
*Automation and remote control*, *25*, 821–837.
- Anbarasi, M., & Saleem Durai, M. (2018). Prediction of protein folding kinetics states using hybrid brainstorm optimization. *International Journal of Computers and Applications*, 1–9. Taylor & Francis.
- Arsuaga-Ríos, M., & Vega-Rodríguez, M. A. (2014). Cost optimization based on brain storming for grid scheduling. *Innovative Computing Technology (INTECH), 2014 Fourth International Conference on* (pp. 31–36). IEEE.
- Arsuaga-Ríos, M., & Vega-Rodríguez, M. A. (2015). Multi-objective energy optimization in grid systems from a brain storming strategy. *Soft Computing*, *19*(11), 3159–3172. Springer.
- Askarzadeh, A. (2014). Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Communications in Nonlinear Science and Numerical Simulation*, *19*(4), 1213–1228. Elsevier.
- Askarzadeh, A., & Rezazadeh, A. (2013). Artificial neural network training using a new efficient optimization algorithm. *Applied Soft Computing*, *13*(2), 1206–1213. Elsevier.
- Avcı, D., & Dogantekin, A. (2016).  
An expert diagnosis system for Parkinson disease based on genetic algorithm-wavelet kernel-extreme learning machine. *Parkinson's disease*, 2016. Hindawi.

- Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.
- Bajeta, M., Sekuli, P., Djukanovi, S., Popovic, T., & PopoviBugarin, V. (2016). Retinalbloodvesselssegmentationusingantcolonyoptimization. *NeuralNetworksandApplications (NEUREL), 2016 13thSymposiumon* (pp. 1–6). IEEE.
- Di Barba, P., Mognaschi, M., & Krawczyk, A. (2017). The biogeography-inspired optimization for the design of coils for nerve stimulation. *Smart Technologies, IEEE EUROCON 2017-17th International Conference on* (pp. 542–545). IEEE.
- Beyer, H.-G., & Schwefel, H.-P. (2002). Evolution strategies—A comprehensive introduction. *Natural computing, 1*(1), 3–52. Springer.
- Binitha, S., Sathya, Ss., & others. (2012a). Asurveyofbioinspiredoptimizationalgorithms. *InternationalJournalofSoftComputingandEngineering, 2*(2), 137–151.
- Binitha, S., Sathya, Ss., & others. (2012b). Asurveyofbioinspiredoptimizationalgorithms. *InternationalJournalofSoftComputingandEngineering, 2*(2), 137–151.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Boser, B., Guyon, I., & Vapnik, V. (1992). Atrainingalgorithmforoptimalmarginclassifiers. *ProceedingsofthefifthannualworkshoponComputationallearningtheory* (pp. 144–152). ACM.
- Brownlee, J. (2011). *Cleveralgorithms: nature-inspiredprogrammingrecipes*. JasonBrownlee.
- Burges, C. J. (1998a). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121–167. Springer.
- Burges, C. J. (1998b). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery, 2*(2), 121–167. Springer.
- Burges, C. J., & Schölkopf, B. (1997). Improving the accuracy and speed of support vector machines. *Advances in neural information processing systems* (pp. 375–381).
- Canayaz, M., & Karci, A. (2016). Cricket behaviour-based evolutionary computation technique in solving engineering optimization problems. *Applied Intelligence, 44*(2), 362–376. Springer.
- Cao, J., Li, M., Wang, H., Huang, L., & Zhao, Y. (2016). AnImprovedBacterialForagingAlgorithmwithcooperativelearningforeradicatingcancercellsusin

- gnanorobots. *Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on* (pp. 1141–1146). IEEE.
- Cao, Y., Chen, W., Cheng, S., Sun, Y., Liu, Q., Li, Y., & Shi, Y. (2017). A simple brain storm optimization algorithm via visualizing confidence intervals. *Asia-Pacific Conference on Simulated Evolution and Learning* (pp. 27–38). Springer.
- Cao, Z., Hei, X., Wang, L., Shi, Y., & Rong, X. (2015). An improved brain storm optimization with differential evolution strategy for applications of ANNs. *Mathematical Problems in Engineering, 2015*. Hindawi.
- Cao, Z., Rong, X., & Du, Z. (2017). An improved brain storm optimization with dynamic clustering strategy. *MATEC web of conferences* (Vol. 95, p. 19002). EDP Sciences.
- Cao, Z., Shi, Y., Rong, X., Liu, B., Du, Z., & Yang, B. (2015). Random grouping brain storm optimization algorithm with a new dynamically changing step size. *International Conference in Swarm Intelligence* (pp. 357–364). Springer.
- De Castro, L. N., & Von Zuben, F. J. (2000). The clonal selection algorithm with engineering applications. *Proceedings of GECCO* (Vol. 2000, pp. 36–39).
- Cervantes-Castillo, A., Leguizamón, G., & Mezura-Montes, E. (2017). Exploring boundary constraints using the modified brain storm optimization algorithm and a boundary search operator: MBSO-BS. *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)* (pp. 1–6). IEEE.
- Cervantes-Castillo, A., & Mezura-Montes, E. (2016). A study of constraint-handling techniques in brain storm optimization. *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3740–3746). IEEE.
- Chandrasekar, R., & Khare, N. (2017). BSFS: Design and Development of Exponential Brain Storm Fuzzy System for Data Classification. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(02), 267–284. World Scientific.
- Chen, J., Cheng, S., Chen, Y., Xie, Y., & Shi, Y. (2015). Enhanced brain storm optimization algorithm for wireless sensor networks deployment. *International Conference in Swarm Intelligence* (pp. 373–381). Springer.

- Chen, J., Wang, J., Cheng, S., & Shi, Y. (2016). Brain storm optimization with agglomerative hierarchical clustering analysis. *International Conference on Swarm Intelligence* (pp. 115–122). Springer.
- Chen, J., Xie, Y., & Ni, J. (2014). Brainstorm optimization model based on uncertainty information. *2014 Tenth International Conference on Computational Intelligence and Security* (pp. 99–103). IEEE.
- Chen, S. (2009). Locust Swarms-A new multi-optima search technique. *2009 IEEE Congress on Evolutionary Computation* (pp. 1745–1752). IEEE.
- Chen, W., Cao, Y., Cheng, S., Sun, Y., Liu, Q., & Li, Y. (2018). Simplex Search-Based Brain Storm Optimization. *IEEE Access*, 6, 75997–76006. IEEE.
- Cheng, S., Lu, H., Song, W., Chen, J., & Shi, Y. (2018). Dynamic Multimodal Optimization Using Brain Storm Optimization Algorithms. *International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 236–245). Springer.
- Cheng, S., Qin, Q., Chen, J., & Shi, Y. (2016). Brain storm optimization algorithm: a review. *Artificial Intelligence Review*, 46(4), 445–458. Springer.
- Cheng, S., Qin, Q., Chen, J., Wang, G.-G., & Shi, Y. (2016). Brain storm optimization in objective space algorithm for multimodal optimization problems. *International Conference in Swarm Intelligence* (pp. 469–478). Springer.
- Cheng, S., Shi, Y., Qin, Q., & Gao, S. (2013). Solution clustering analysis in brain storm optimization algorithm. *Swarm Intelligence (SIS), 2013 IEEE Symposium on* (pp. 111–118). IEEE.
- Cheng, S., Shi, Y., Qin, Q., Ting, T. O., & Bai, R. (2014). Maintaining population diversity in brain storm optimization algorithm. *IEEE Congress on Evolutionary Computation* (pp. 3230–3237).
- Cheng, S., Sun, Y., Chen, J., Qin, Q., Chu, X., Lei, X., & Shi, Y. (2017). A comprehensive survey of brain storm optimization algorithms. *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1637–1644). IEEE.
- Chitsaz, H., Amjady, N., & Zareipour, H. (2015). Wind power forecast using wavelet neural network trained by improved Clonal selection algorithm. *Energy conversion and Management*, 89, 588–598. Elsevier.

- Chu, S.-C., Tsai, P.-W., & Pan, J.-S. (2006). Cat swarm optimization. *Pacific Rim international conference on artificial intelligence* (pp. 854–858). Springer.
- Chu, X., Chen, J., Cai, F., Chen, C., & Niu, B. (2017). Augmented Brain Storm Optimization with Mutation Strategies. *Asia-Pacific Conference on Simulated Evolution and Learning* (pp. 949–959). Springer.
- Conforth, M., & Meng, Y. (2008). Toward Evolving Neural Networks using Bio-Inspired Algorithms. *IC-AI* (pp. 413–419).
- Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, *40*(16), 6374–6384. Elsevier.
- Cutello, V., Narzisi, G., Nicosia, G., & Pavone, M. (2005). Clonal selection algorithms: a comparative case study using effective mutation potentials. *International Conference on Artificial Immune Systems* (pp. 13–28). Springer.
- Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. *Foundations of Computational Intelligence Volume 3* (pp. 23–55). Springer.
- Dash, S., Joshi, D., & Trivedi, G. (2018). Multiobjective analog/RF circuit sizing using an improved brain storm optimization algorithm. *Memetic Computing*, *10*(4), 423–440. Springer.
- Dhiman, G., & Kumar, V. (2017). Spottedhyenaoptimizer: Anovelbio-inspiredbasedmetaheuristictechniqueforengineeringapplications. *AdvancesinEngineeringSoftware*, *114*, 48–70. Elsevier.
- Dhiman, G., & Kumar, V. (2018). Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge-Based Systems*, *159*, 20–50. Elsevier.
- Dolicanin, E., Fetahovic, I., Tuba, E., Capor-Hrosik, R., & Tuba, M. (2018). Unmanned combat aerial vehicle path planning by brain storm optimization algorithm. *Studies in Informatics and Control*, *27*(1), 15–24.

- Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (Vol. 2, pp. 1470–1477). IEEE.
- Duan, H., & Li, C. (2015). Quantum-behaved brain storm optimization approach to solving Loney's solenoid problem. *IEEE Transactions on Magnetics*, 51(1), 1–7. IEEE.
- Duan, H., Li, S., & Shi, Y. (2013). Predator–prey brain storm optimization for DC brushless motor. *IEEE Transactions on Magnetics*, 49(10), 5336–5340. IEEE.
- Duan, H., Zhang, D., Shi, Y., & Deng, Y. (2018). Close formation flight of swarm unmanned aerial vehicles via metric-distance brain storm optimization. *Memetic Computing*, 10(4), 369–381. Springer.
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). Ieee.
- Ebrahimi, A., & Khomehchi, E. (2016). Sperm whale algorithm: an effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering*, 29, 211–222. Elsevier.
- Eesa, A. S., Brifcani, A. M. A., & Orman, Z. (2013). Cuttlefish algorithm-a novel bio-inspired optimization algorithm. *International Journal of Scientific & Engineering Research*, 4(9), 1978–1986.
- El-Abd, M. (2016a). Brain storm optimization algorithm with re-initialized ideas and adaptive step size. *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2682–2686). IEEE.
- El-Abd, M. (2016b). Cooperative coevolution using the brain storm optimization algorithm. *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on* (pp. 1–7). IEEE.
- El-Abd, M. (2017). Global-best brain storm optimization algorithm. *Swarm and Evolutionary Computation*, 37, 27–44. Elsevier.
- ElMajdoui, M., Bougrine, S., Rbouch, I., & ElImrani, A. (2017). A comparative study of the EEG signals big optimization problem using evolutionary, swarm and memetic computational algorithms.

- Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1357–1364). ACM.
- Faiz, O., Blackburn, S., & Moffat, D. (2011). *Anatomy at a Glance* (Vol. 66). John Wiley & Sons.
- Ferster, D., & Spruston, N. (1995). *Cracking the neuronal code*. *Science* (Vol. 270, pp. 756–757). American Association for the Advancement of Science.
- Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister. (2013a). A brief review of nature-inspired algorithms for optimization.
- Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013b). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.
- Fogel, D. (1993). Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and systems*, 24(1), 27–36. Taylor & Francis.
- Fogel, D. B., & Fogel, L. J. (1995). An introduction to evolutionary programming. *European Conference on Artificial Evolution* (pp. 21–33). Springer.
- Fu, Y., Wang, H., & Huang, M. (2018). Integrated scheduling for a distributed manufacturing system: a stochastic multi-objective model. *Enterprise Information Systems*, 1–17. Taylor & Francis.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193–202. Springer.
- Gandomi, A., & Alavi, A. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831–4845. Elsevier.
- Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1), 17–35. Springer.
- Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, 22(6), 1239–1255. Springer.
- Ganesh, S., & Kanimozhi, R. (2018). Multi Objective Approach for Capacitor Placement Based on Load Balancing Index in Radial Distribution System. *Journal of Computational and Theoretical Nanoscience*, 15(1), 282–288. American Scientific Publishers.



- García-Ródenas, R., Linares, L. J., & López-Gómez, J. A. (2017). A cooperative brain storm optimization algorithm. *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 838–845). IEEE.
- Garro, B. A., Rodríguez, K., & Vázquez, R. A. (2016). Classification of DNA microarrays using artificial neural networks and ABC algorithm. *Applied Soft Computing*, 38, 548–560. Elsevier.
- Gheraibia, Y., & Moussaoui, A. (2013). Penguins search optimization algorithm (PeSOA). *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 222–231). Springer.
- Gnanasekaran, N., Chandramohan, S., Kumar, P. S., & Imran, A. M. (2016). Optimal placement of capacitors in radial distribution system using shark smell optimization algorithm. *Ain Shams Engineering Journal*, 7(2), 907–916. Elsevier.
- Goel, S., Sharma, A., & Panchal, V. (2014). Performance analysis of bio-inspired techniques. *Proceedings of the Third International Conference on Soft Computing for Problem Solving* (pp. 831–844). Springer.
- Gray, H. (1878). *Anatomy of the human body* (Vol. 8). Lea & Febiger.
- Gudise, V. G., & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)* (pp. 110–117). IEEE.
- Guo, Q., Xue, Z., Zhang, L., Lu, X., Yin, Y., & Huang, C. (2018). Wind Turbine Unit Power Prediction Based on Wavelet Neural Network Optimized by Brain Storm Optimization Algorithm. *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)* (pp. 664–669). IEEE.
- Guo, X., Wu, Y., & Xie, L. (2014). Modified brain storm optimization algorithm for multimodal optimization. *International Conference in Swarm Intelligence* (pp. 340–351). Springer.
- Guo, X., Wu, Y., Xie, L., Cheng, S., & Xin, J. (2015). An adaptive brain storm optimization algorithm for multiobjective optimization problems. *International conference in swarm intelligence* (pp. 365–372). Springer.

- Gupta, U., Swarnalatha, P., & Chharia, P. (2016). Speech guided feature extraction and brain storm optimization to cluster objects using fuzzy logic. *IIOABJ*, 7, 193–202.
- Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, Prentice Hall.
- Holland, J. H. (1984). Genetic algorithms and adaptation. *Adaptive Control of Ill-Defined Systems* (pp. 317–333). Springer.
- Hong, Y., Hao, Z., Mai, G., Huang, H., & Kumar Sangaiah, A. (2018). Causal Discovery Combining K2 with Brain Storm Optimization Algorithm. *Molecules*, 23(7), 1729. Multidisciplinary Digital Publishing Institute.
- Hsu, C.-W., Chang, C.-C., Lin, C.-J., & others. (2003). A practical guide to support vector classification. Taipei.
- Hua, Z., Chen, J., & Xie, Y. (2016). Brain storm optimization with discrete particle swarm optimization for TSP. *Computational Intelligence and Security (CIS), 2016 12th International Conference on* (pp. 190–193). IEEE.
- Huang, J. S., Wu, Q., & Chen, Q. (2015). Design Swapping Encryption System for Video Streaming with Brain Storm Optimization. *Bio-Inspired Computing-Theories and Applications* (pp. 173–180). Springer.
- I. Vlahavas, I. S. P. Kefalas, F. Kokkoras. (2006). *Artificial Intelligence*. V. Giurdas Publications.
- Ibrahim, R. A., Elaziz, M. A., Ewees, A. A., Selim, I. M., & Lu, S. (2018). Galaxy images classification using hybrid brain storm optimization with moth flame optimization. *Journal of Astronomical Telescopes, Instruments, and Systems*, 4(3), 038001. International Society for Optics and Photonics.
- Jaddi, N. S., Abdullah, S., & Hamdan, A. R. (2015). Multi-population cooperative bat algorithm-based optimization of artificial neural network model. *Information Sciences*, 294, 628–644. Elsevier.
- Jadhav, H., Sharma, U., Patel, J., & Roy, R. (2012). Brain storm optimization algorithm based economic dispatch considering wind power. *Power and Energy (PECon), 2012 IEEE International Conference on* (pp. 588–593). IEEE.
- Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3), 31–44. IEEE.

- Jiang, P., & Li, P. (2016). Research and application of a new hybrid wind speed forecasting model on BSO algorithm. *Journal of Energy Engineering*, 143(1), 04016019. American Society of Civil Engineers.
- Jiang, P., Qin, S., Wu, J., & Sun, B. (2015). Time series analysis and forecasting for wind speeds using support vector regression coupled with artificial intelligent algorithms. *Mathematical Problems in Engineering*, 2015. Hindawi.
- Jin, C., & Qin, A. K. (2017). A GPU-based implementation of brain storm optimization. *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2698–2705). IEEE.
- Jordehi, A. R. (2015). Brainstorm optimisation algorithm (BSOA): An efficient algorithm for finding optimal location and setting of FACTS devices in electric power systems. *International Journal of Electrical Power & Energy Systems*, 69, 48–57. Elsevier.
- Karaboga, D., Akay, B., & Ozturk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. *International conference on modeling decisions for artificial intelligence* (pp. 318–329). Springer.
- Karaboga, D., & Aslan, S. (2016). A discrete artificial bee colony algorithm for detecting transcription factor binding sites in DNA sequences. *Genet. Mol. Res*, 15(2), 1–11.
- Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 42(1), 21–57. Springer.
- Kawaguchi, K. (2001a). A multithreaded software model for backpropagation neural network applications.
- Kawaguchi, K. (2001b). A multithreaded software model for backpropagation neural network applications.
- Ke, L. (2018). A brain storm optimization approach for the cumulative capacitated vehicle routing problem. *Memetic Computing*, 1–11. Springer.
- Koutny, T. (2016). Using meta-differential evolution to enhance calculation of a continuous blood glucose level. *Computer methods and programs in biomedicine*, 133, 45–54. Elsevier.

- Kowalski, P. A., & Lukasik, S. (2016). Training neural networks with krill herd algorithm. *Neural Processing Letters*, 44(1), 5–17. Springer.
- Koza, J. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2), 87–112. Springer.
- Krishnanand, K., & Ghose, D. (2009). Glowworm swarm optimisation: a new method for optimising multi-modal functions. *International Journal of Computational Intelligence Studies*, 1(1), 93–119. Inderscience Publishers.
- Lei, X., Li, H., Zhang, A., & Wu, F.-X. (2017). iOPTICS-GSO for identifying protein complexes from dynamic PPI networks. *BMC medical genomics*, 10(5), 80. BioMed Central.
- Leung, F. H.-F., Lam, H.-K., Ling, S.-H., & Tam, P. K.-S. (2003). Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural networks*, 14(1), 79–88. IEEE.
- Li, C., & Duan, H. (2015). Information granulation-based fuzzy RBFNN for image fusion based on chaotic brain storm optimization. *Optik-International Journal for Light and Electron Optics*, 126(15-16), 1400–1406. Elsevier.
- Li, C., Hu, D., Song, Z., Yang, F., Luo, Z., Fan, J., & Liu, P. (2018). A Vector Grouping Learning Brain Storm Optimization Algorithm for Global Optimization Problems. *IEEE Access*, 6, 78193–78213. IEEE.
- Li, D., Luo, L., Zhang, W., Liu, F., & Luo, F. (2016). A genetic algorithm-based weighted ensemble method for predicting transposon-derived piRNAs. *BMC bioinformatics*, 17(1), 329. BioMed Central.
- Li, G., Jin, Y., Akram, M., Chen, X., & Ji, J. (2018). Application of bio-inspired algorithms in maximum power point tracking for PV systems under partial shading conditions – a review. *Renewable and Sustainable Energy Reviews*, 81, 840–873. Elsevier.
- Li, H.-Z., Guo, S., Li, C.-J., & Sun, J.-Q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowledge-Based Systems*, 37, 378–387. Elsevier.

- Li, J., & Duan, H. (2015). Simplified brain storm optimization approach to control parameter optimization in F/A-18 automatic carrier landing system. *Aerospace Science and Technology*, 42, 187–195. Elsevier.
- Li, L., Zhang, F., Chu, X., & Niu, B. (2016). Modified brain storm optimization algorithms based on topology structures. *International Conference on Swarm Intelligence* (pp. 408–415). Springer.
- Li, X., Zhang, J., & Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, 24(7-8), 1867–1877. Springer.
- Liang, J., Wang, P., Yue, C., Yu, K., Li, Z., & Qu, B. (2018). Multi-objective Brainstorm Optimization Algorithm for Sparse Optimization. *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8). IEEE.
- Liu, J., Ji, J., Zhang, A., & Liang, P. (2016). An ant colony optimization algorithm for learning brain effective connectivity network from fMRI data. *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on* (pp. 360–367). IEEE.
- Ma, X., Jin, Y., & Dong, Q. (2017). A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting. *Applied Soft Computing*, 54, 296–312. Elsevier.
- Ma, X., & Xian, N. (2016). A Re-estimation Brain Storm Optimization to Train Hidden Markov Model for Transcription Factor Binding Site Analysis. *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on* (pp. 134–139). IEEE.
- Madheswari, K., Venkateswaran, N., & Sowmiya, V. (2016). Visible and thermal image fusion using curvelet transform and brain storm optimization. *Region 10 Conference (TENCON), 2016 IEEE* (pp. 2826–2829). IEEE.
- Maftciu-Scai, L. O. (2015). A new approach for solving equations systems inspired from brainstorming. *Int J New Comput Archit Appl*, 5(1), 10–18.

- Maftciu-Scai, L. O., Maftciu-Scai, E., & Voina, T. (2017). Bandwidths Optimization on Sparse Matrices Using Brain Storm Optimization. *2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (pp. 219–224). IEEE.
- Maia, R. D., Castro, L. N. de, & Caminhas, W. M. (2012). Bee colonies as model for multimodal continuous optimization: The OptBees algorithm. *2012 IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE.
- Margain, L., Ochoa, A., Almaguer, L. M., & Velázquez, R. (2017). Model on Oil Platform Using Brain Storm Optimization Algorithm. *International Conference on Health Information Science* (pp. 311–320). Springer.
- Marinakis, Y., & Marinaki, M. (2014). A bumble bees mating optimization algorithm for the open vehicle routing problem. *Swarm and Evolutionary Computation*, *15*, 80–94. Elsevier.
- Mehrabian, A., & Lucas, C. (2006). Anovelnumericaloptimizationalgorithminspiredfromweedcolonization. *Ecologicalinformatics*, *1*(4), 355–366. Elsevier.
- Mehrnam, A., Nasrabadi, A., Ghodousi, M., Mohammadian, A., & Torabi, S. (2017). A new approach to analyze data from EEG-based concealed face recognition system. *International Journal of Psychophysiology*, *116*, 1–8. Elsevier.
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, *89*, 228–249. Elsevier.
- Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51–67. Elsevier.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46–61. Elsevier.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Niu, B., Liu, J., Liu, J., & Yang, C. (2016). Brain storm optimization for portfolio optimization. *International Conference in Swarm Intelligence* (pp. 416–423). Springer.

- Nucamendi-Guillén, S., Dávila, D., Camacho-Vallejo, J.-F., & González-Ramírez, R. G. (2018). A discrete bilevel brain storm algorithm for solving a sales territory design problem: a case study. *Memetic Computing*, *10*(4), 441–458. Springer.
- Oliveira Batista, J. de, Rodrigues, R. B., & Varejão, F. M. (2017). Soft computing classifier ensemble for fault diagnosis. *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)* (pp. 1348–1353). IEEE.
- Pal, P. S., Kar, R., Mandal, D., & Ghoshal, S. P. (2016). Identification of NARMAX Hammerstein models with performance assessment using brain storm optimization algorithm. *International Journal of Adaptive Control and Signal Processing*, *30*(7), 1043–1070. Wiley Online Library.
- Pal, P. S., Kar, R., Mandal, D., & Ghoshal, S. P. (2017). Parametric identification with performance assessment of wiener systems using brain storm optimization algorithm. *Circuits, Systems, and Signal Processing*, *36*(8), 3143–3181. Springer.
- Pan, W.-T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, *26*, 69–74. Elsevier.
- Pérez-Cruz, J. H., Portilla-Flores, E. A., Niño-Suárez, P. A., & Rivera-Blas, R. (2017). Design of a nonlinear controller and its intelligent optimization for exponential synchronization of a new chaotic system. *Optik-International Journal for Light and Electron Optics*, *130*, 201–212. Elsevier.
- Qiao, Y., Huang, Y., & Gao, Y. (2018). The Global Convergence Analysis of Brain Storm Optimization. *NeuroQuantology*, *16*(6).
- Qiu, H., & Duan, H. (2014). Receding horizon control for multiple UAV formation flight based on modified brain storm optimization. *Nonlinear dynamics*, *78*(3), 1973–1988. Springer.
- Qiu, H., Duan, H., Zhou, Z., Hu, X., & Shi, Y. (2017). Chaotic predator-prey brain storm optimization for continuous optimization problems. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–7). IEEE.
- Rafael Gonzalez, R. W. (2008). *Digital Image Processing*. Pearson Education.

- Raj, S., Ray, K., & Shankar, O. (2016).  
 Cardiac arrhythmia beat classification using DOST and PSO tuned SVM.  
*Computer methods and programs in biomedicine*, 136, 163–177. Elsevier.
- Ramadan, R., & Khedr, A. (2016). Brain storming algorithm for coverage and connectivity problem in wireless sensor network. *Communication, Management and Information Technology: International Conference on Communication, Management and Information Technology (ICCMIT 2016, Cosenza, Italy, 26-29 April 2016)* (p. 371). CRC Press.
- Ramanand, K., Krishnanand, K., Panigrahi, B. K., & Mallick, M. K. (2012). Brain storming incorporated teaching–learning–based algorithm with application to electric power dispatch. *International Conference on Swarm, Evolutionary, and Memetic Computing* (pp. 476–483). Springer.
- Ramyachitra, D., & Veeralakshmi, V. (2017).  
 Bacterial Foraging Optimization for protein structure prediction using FCC & HP energy model.  
*Gene Reports*, 7, 43–49. Elsevier.
- Ravi, C., & Khare, N. (2018). BGFS: Design and development of brain genetic fuzzy system for data classification. *Journal of Intelligent Systems*, 27(2), 231–247. De Gruyter.
- Revathi, S. T., Ramaraj, N., & Chithra, S. (n.d.). Brain storm-based Whale Optimization Algorithm for privacy-protected data publishing in cloud computing. *Cluster Computing*, 1–10. Springer.
- Roy, R., & Anuradha, J. (2015). A modified brain storm optimization for clustering using hard c-means. *Research in Computational Intelligence and Communication Networks (ICRCICN), 2015 IEEE International Conference on* (pp. 202–207). IEEE.
- Sai, V.-O., Shieh, C.-S., Lin, Y.-C., Horng, M.-F., Nguyen, T.-T., Le, Q.-D., & Jiang, J.-Y. (2016). Comparative study on recent development of heuristic optimization methods. *2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)* (pp. 68–71). IEEE.
- Sambandam, B., Josephine, M., & Jeyabalaraja, V. (2017).  
 Segmentation of Optic Disc and Optic Cup Using Krill Herd Algorithm for the Assessment of Glaucoma. *International Journal of Applied Engineering Research*, 12(21), 10938–10944.



- Sato, M., & Fukuyama, Y. (2018). Total Optimization of Smart City by Brain Storm Optimization. *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)* (pp. 1364–1369). IEEE.
- Sayed, G., Soliman, M., & Hassanien, A. (2018). Modified optimal foraging algorithm for parameters optimization of support vector machine. *International Conference on Advanced Machine Learning Technologies and Applications* (pp. 23–32). Springer.
- Sheta, A., Braik, M., & Al-Hiary, H. (2019). Modeling the Tennessee Eastman chemical process reactor using bio-inspired feedforward neural network (BI-FF-NN). *The International Journal of Advanced Manufacturing Technology*, 1–22. Springer.
- Shi, Y. (2011). Brain storm optimization algorithm. *International Conference in Swarm Intelligence* (pp. 303–309). Springer.
- Shi, Y. (2015). Brain storm optimization algorithm in objective space. *2015 IEEE Congress on evolutionary computation (CEC)* (pp. 1227–1234). IEEE.
- Shi, Y., & Eberhart, R. (1999). Empirical study of particle swarm optimization. *Evolutionary computation, 1999. CEC 99. Proceedings of the 1999 congress on* (Vol. 3, pp. 1945–1950). IEEE.
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702–713. IEEE.
- Simon, & Haykin. (2009). *Neural networks and learning machines* (Vol. 3). Pearson Upper Saddle River.
- Sivaranjani, R., Roomi, S. M. M., & Senthilarasi, M. (2019). Speckle noise removal in SAR images using Multi-Objective PSO (MOPSO) algorithm. *Applied Soft Computing*, 76, 671–681. Elsevier.
- Snell, R. S. (2010). *Clinical neuroanatomy*. Lippincott Williams & Wilkins.
- Socha, K., & Blum, C. (2007). An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. *Neural Computing and Applications*, 16(3), 235–247. Springer.

- Soyinka, O. K., & Duan, H. (2016). Optimal Impulsive Thrust Trajectories for Satellite Formation via Improved Brainstorm Optimization. *International Conference in Swarm Intelligence* (pp. 491–499). Springer.
- Storn, R., & Price, K. (1997). Differential evolution— a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341–359. Springer.
- Stufflebeam, R. (2008). Neurons, synapses, action potentials, and neurotransmission. *Consortium on Cognitive Science Instruction*.
- Subudhi, B., & Jena, D. (2011). A differential evolution based neural network approach to nonlinear system identification. *Applied Soft Computing*, *11*(1), 861–871. Elsevier.
- Sun, C., Duan, H., & Shi, Y. (2013). Optimal satellite formation reconfiguration based on closed-loop brain storm optimization. *IEEE Computational Intelligence Magazine*, *8*(4), 39–51. IEEE.
- Sun, Y. (2014). A hybrid approach by integrating brain storm optimization algorithm with grey neural network for stock index forecasting. *Abstract and Applied Analysis* (Vol. 2014). Hindawi.
- Sun, Y., Jin, Y., & Wang, D. (2018). Brain Storm Optimization with Multi-population Based Ensemble of Creating Operations. *International Conference on Swarm Intelligence* (pp. 374–383). Springer.
- Sveshnikova, A. N., Panteleev, M. A., Dreval, A. V., Shestakova, T. P., Medvedev, O. S., & Dreval, O. A. (2017). Theoretical evaluation of the parameters of glucose metabolism on the basis of continuous glycemia monitoring data using mathematical modeling. *Biophysics (Russian Federation)*, *62*(5), 842–847. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85038438460&doi=10.1134%2fS0006350917050220&partnerID=40&md5=22e948c5e5440f45615b1d8299d3239b>
- Tahmassebi, A., Gandomi, A. H., McCann, I., Schulte, M. H., Schmaal, L., Goudriaan, A. E., & Meyer-Baese, A. (2017). An evolutionary approach for fMRI big data classification. *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 1029–1036). IEEE.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

- Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2018). Deep learning in spiking neural networks. *Neural Networks*. Elsevier.
- Ting, T. O., & Shi, Y. (2016). Parameter estimation of vertical two-layer soil model via brain storm optimization algorithm. *International Conference on Swarm Intelligence* (pp. 500–511). Springer.
- Tsai, H.-C., & Lin, Y.-H. (2011). Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior. *Applied Soft Computing*, *11*(8), 5367–5374. Elsevier.
- Tuba, E., Capor-Hrosik, R., Alihodzic, A., & Tuba, M. (2017). Drone placement for optimal coverage by brain storm optimization algorithm. *International Conference on Health Information Science* (pp. 167–176). Springer.
- Tuba, E., Dolicanin, E., & Tuba, M. (2017). Chaotic brain storm optimization algorithm. *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 551–559). Springer.
- Tuba, E., Simian, D., Dolicanin, E., Jovanovic, R., & Tuba, M. (2018). Energy Efficient Sink Placement in Wireless Sensor Networks by Brain Storm Optimization Algorithm. *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)* (pp. 718–723). IEEE.
- Tuba, E., Strumberger, I., Zivkovic, D., Bacanin, N., & Tuba, M. (2018). Mobile Robot Path Planning by Improved Brain Storm Optimization Algorithm. *2018 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8). IEEE.
- Ulagammai, M., Venkatesh, P., Kannan, P., & Padhy, N. P. (2007). Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting. *Neurocomputing*, *70*(16-18), 2659–2667. Elsevier.
- Vapnik, V. N. (1995). Estimation of dependences based on empirical data. 1982. NY: Springer-Verlag.
- Verma, D., & Dubey, S. (2017). Fuzzy brain storm optimization and adaptive thresholding for multimodal vein-based recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, *31*(05), 1756007. World Scientific.

- Wang, D., Luo, H., Grunder, O., Lin, Y., & Guo, H. (2017). Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm. *Applied Energy*, *190*, 390–407. Elsevier.
- Wang, G.-G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural computing and applications*, *31*(7), 1995–2014. Springer.
- Wang, G.-G., Hao, G.-S., Cheng, S., Shi, Y., & Cui, Z. (2017). An improved brain storm optimization algorithm based on graph theory. *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 509–515). IEEE.
- Wang, H., Liu, J., Yi, W., Niu, B., & Baek, J. (2017). An Improved Brain Storm Optimization with Learning Strategy. *International Conference on Swarm Intelligence* (pp. 511–518). Springer.
- Wang, J., Chen, J., Xue, X., & Du, J. (2018). Search strategies investigation in brain storm optimization. *Memetic Computing*, *10*(4), 397–409. Springer.
- Wang, J., Hou, R., Wang, C., & Shen, L. (2016). Improved v-support vector regression model based on variable selection and brain storm optimization for stock price forecasting. *Applied Soft Computing*, *49*, 164–178. Elsevier.
- Ward, J., & Linden, R. (2017). *Physiology at a Glance*. John Wiley & Sons.
- Wei, M., & Shi, Y. (2015). Brain storm optimization algorithms for optimal coverage of wireless sensor networks. *Technologies and Applications of Artificial Intelligence (TAAI), 2015 Conference on* (pp. 120–127). IEEE.
- Wu, T., Yao, M., & Yang, J. (2016). Dolphin swarm algorithm. *Frontiers of Information Technology & Electronic Engineering*, *17*(8), 717–729. Springer.
- Wu, X.-L., Zhang, Z.-Q., & Li, J.-Q. (2017). A brain storm optimization algorithm integrating diversity and discussion mechanism for solving discrete production scheduling problem. *Kongzhi yu Juece/Control and Decision*, *32*, 1583–1590.
- Wu, Y., Wang, X., Fu, Y., & Xu, Y. (2017). Difference Brain Storm Optimization for Combined Heat and Power Economic Dispatch. *International Conference in Swarm Intelligence* (pp. 519–527). Springer.

- Wu, Y., Xie, L., & Liu, Q. (2016). Multi-objective brain storm optimization based on estimating in knee region and clustering in objective-space. *International Conference on Swarm Intelligence* (pp. 479–490). Springer.
- Xie, L., & Wu, Y. (2014). A modified multi-objective optimization based on brain storm optimization algorithm. *International conference in swarm intelligence* (pp. 328–339). Springer.
- Xiong, G., Shi, D., Zhang, J., & Zhang, Y. (2018). A binary coded brain storm optimization for fault section diagnosis of power systems. *Electric Power Systems Research*, 163, 441–451. Elsevier.
- Xu, Y., Wu, Y., Fu, Y., Wang, X., & Lu, A. (2018). Discrete brain storm optimization algorithm based on prior knowledge for traveling salesman problems. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 2740–2745). IEEE.
- Xue, J., Wu, Y., Shi, Y., & Cheng, S. (2012). Brain storm optimization algorithm for multi-objective optimization problems. *International Conference in Swarm Intelligence* (pp. 513–519). Springer.
- Xue, Y., Tang, T., & Ma, T. (2016). Classification based on brain storm optimization algorithm. *International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 371–376). Springer.
- Yadav, P. (2016). Case retrieval algorithm using similarity measure and adaptive fractional brain storm optimization for health informaticians. *Arabian Journal for Science and Engineering*, 41(3), 829–840. Springer.
- Yadav, P. (2018). Cluster based-image descriptors and fractional hybrid optimization for medical image retrieval. *Cluster Computing*, 1–15. Springer.
- Yan, X., Hao, Z., Huang, H., & Li, G. (2016). Human-computer cooperative brain storm optimization algorithm for the two-echelon vehicle routing problem. *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2676–2681). IEEE.
- Yan, X., Zhu, Z., Wu, Q., Gong, W., & Wang, L. (2018). Elastic parameter inversion problem based on brain storm optimization algorithm. *Memetic Computing*, 1–11. Springer.

- Yang, C., & Simon, D. (2005). *A new particleswarm optimization technique*.  
*A new particleswarm optimization technique* (pp. 164–169). IEEE.
- Yang, G., Wang, B., Zheng, X., Zhou, C., & Zhang, Q. (2017).  
 IWO Algorithm Based on Niche Crowding for DNA Sequence Design. *Interdisciplinary Sciences: Computational Life Sciences*, 9(3), 341–349. Springer.
- Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. *International symposium on stochastic algorithms* (pp. 169–178). Springer.
- Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm.  
*Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65–74). Springer.
- Yang, Y., Shi, Y., & Xia, S. (2015). Advanced discussion mechanism-based brain storm optimization algorithm. *Soft computing*, 19(10), 2997–3007. Springer.
- Yang, Z., & Shi, Y. (2015). Brain storm optimization with chaotic operation. *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)* (pp. 111–115). IEEE.
- Yao, K.-H., Jiang, J.-R., Tsai, C.-H., & Wu, Z.-S. (2017).  
 Evolutionary Beamforming Optimization for Radio Frequency Charging in Wireless Rechargeable Sensor Networks. *Sensors*, 17(8), 1918. Multidisciplinary Digital Publishing Institute.
- Yazdani, M., & Jolai, F. (2016). Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, 3(1), 24–36. Elsevier.
- Yu, Y., Gao, S., Wang, Y., Cheng, J., & Todo, Y. (2018). ASBSO: An Improved Brain Storm Optimization With Flexible Search Length and Memory-Based Selection. *IEEE Access*, 6, 36977–36994. IEEE.
- Yu, Y., Wu, L., Yu, H., Li, S., Wang, S., & Gao, S. (2017). Brain storm optimization with adaptive search radius for optimization. *Progress in Informatics and Computing (PIC), 2017 International Conference on* (pp. 394–398). IEEE.
- Yusiong, J. P. T. (2012). Optimizing artificial neural networks using cat swarm optimization algorithm. *International Journal of Intelligent Systems and Applications*, 5(1), 69. Modern Education and Computer Science Press.

- Zaharis, Z. D., Skeberis, C., Xenos, T. D., Lazaridis, P. I., & Cosmas, J. (2013). Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data. *IEEE Transactions on Broadcasting*, 59(3), 455–460. IEEE.
- Zamani, M., & Kremer, S. (2016). Protein secondary structure prediction through a novel framework of secondary structure transitions sites and new encoding schemes (pp. 1–7). IEEE.
- Zhan, Z., Chen, W., Lin, Y., Gong, Y., Li, Y., & Zhang, J. (2013). Parameter investigation in brain storm optimization. *Swarm Intelligence (SIS), 2013 IEEE Symposium on* (pp. 103–110). IEEE.
- Zhan, Z., Zhang, J., Shi, Y., & Liu, H. (2012). A modified brain storm optimization. *Evolutionary Computation (CEC), 2012 IEEE Congress on* (pp. 1–8). IEEE.
- Zhang, G.-W., Zhan, Z.-H., Du, K.-J., & Chen, W.-N. (2014). Normalization group brain storm optimization for power electronic circuit optimization. *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation* (pp. 183–184). ACM.
- Zhang, J.-R., Zhang, J., Lok, T.-M., & Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Applied mathematics and computation*, 185(2), 1026–1037. Elsevier.
- Zhang, W., & Wang, S. (2017). An integrated framework for identifying mutated driver pathway and cancer progression. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (1), 1–1. IEEE.
- Zhang, X., Zhang, Y., Gao, H., & He, C. (2018). A Wrapper Feature Selection Algorithm Based on Brain Storm Optimization. *International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 308–315). Springer.
- Zhao, J., & Feng, H. (2016). Estimating isoform abundance by Particle Swarm Optimization. *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on* (pp. 161–166). IEEE.

- Zheng, X., Lei, Y., Gong, M., & Tang, Z. (2016). Multifactorial brain storm optimization algorithm. *International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 47–53). Springer.
- Zhou, D., Shi, Y., & Cheng, S. (2012). Brain storm optimization algorithm with modified step-size and individual generation. *International Conference in Swarm Intelligence* (pp. 243–252). Springer.
- Zhou, H. J., Jiang, M. Y., & Ben, X. Y. (2014). Niche brain storm optimization algorithm for multi-peak function optimization. *Advanced Materials Research* (Vol. 989, pp. 1626–1630). Trans Tech Publ.
- Zhou, J., Wu, K., & Rao, C. V. (2016). Evolutionary engineering of *Geobacillus thermoglucosidasius* for improved ethanol production. *Biotechnology and bioengineering*, 113(10), 2156–2167. Wiley Online Library.
- Zhou, Y., Chen, X., & Zhou, G. (2016). An improved monkey algorithm for a 0-1 knapsack problem. *Applied Soft Computing*, 38, 817–830. Elsevier.
- Zhou, Z., Duan, H., & Shi, Y. (2016). Convergence analysis of brain storm optimization algorithm. *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 3747–3752). IEEE.
- Zhu, H., & Shi, Y. (2015). Brain storm optimization algorithms with k-medians clustering algorithms. *Advanced Computational Intelligence (ICACI), 2015 Seventh International Conference on* (pp. 107–110). IEEE.
- Zhu, H., & Shi, Y. (2016). Brain storm optimization algorithm for full area coverage of wireless sensor networks. *Advanced Computational Intelligence (ICACI), 2016 Eighth International Conference on* (pp. 14–20). IEEE.
- Βασιλειος Γ.Καμπουραζος, Γ. Α. Π. (2015). *Εισαγωγή στην υπολογιστική νοημοσύνη*.
- Βλαχabas, Σ. Κεφαλας Βασιλειαδης Κοκκορας. (2006). *Τεχνητη Νοημοσύνη*. Εκδοσεις Πανεπιστημιου Μακεδονιας.
- Διαμανταρας. (2007). *Τεχνητα Νευρωνικα Δικτυα. Κλειδαριθμος*.
- Παπαμαρκος, Ν. Η. (2015). *Ψηφιακη επεξεργασια & αναλυση εικονας*. (3η Εκδοση, Ed.).





