



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Εφαρμογές του αλγόριθμου «Βελτιστοποίησης
Σμήνους Σωματιδίων»
σε προβλήματα ταξινόμησης και ομαδοποίησης**

Μιχάλης Π. Χριστόπουλος

Επιβλέπωντας: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΑΘΗΝΑ

Ιανουάριος 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Εφαρμογές του αλγόριθμου «Βελτιστοποίησης Σμήνους Σωματιδίων» σε
προβλήματα ταξινόμησης και ομαδοποίησης**

Μιχάλης Π. Χριστόπουλος
A.M.: M1260

ΕΠΙΒΛΕΠΟΝΤΑΣ: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής

ΑΞΙΟΛΟΓΗΤΗΣ: Βασίλειος Ζησιμόπουλος, Καθηγητής

ΑΘΗΝΑ

Ιανουάριος 2015

ΠΕΡΙΛΗΨΗ

Στην εργασία μελετάται και αναλύεται η χρήση του αλγόριθμου βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization - PSO) σε προβλήματα ταξινόμησης και ομαδοποίησης (επιβλεπόμενης και μη-επιβλεπόμενης μάθησης).

Ο αλγόριθμος PSO μπορεί να χρησιμοποιηθεί σε ένα πρόβλημα ταξινόμησης ή ομαδοποίησης μετατρέποντάς το σε πρόβλημα βελτιστοποίησης των θέσεων των κέντρων των ομάδων. Το κύριο πλεονέκτημά του είναι ότι δεν εξαρτάται από τις τιμές αρχικοποίησης και πραγματοποιεί αναζήτηση σε ευρύτερο χώρο σε σχέση με άλλους αλγορίθμους, με αποτέλεσμα να μην καταλήγει εύκολα σε τοπικά βέλτιστες λύσεις.

Μελετώνται επίσης παραλλαγές του PSO οι οποίες εφαρμόζονται για να ξεπεραστούν κάποια προβλήματα ή για να αυξηθεί η αποτελεσματικότητά του. Προτείνεται μια μέθοδος βελτίωσης της αποτελεσματικότητας του PSO με τη χρήση γραμμικής αύξησης της επιρροής της προσωπικής παραμέτρου και ταυτόχρονα γραμμικής μείωσης της επιρροής της κοινωνικής παραμέτρου.

Τέλος, γίνεται πειραματική αξιολόγησή του αλγορίθμου, με και χωρίς την προτεινόμενη βελτίωση, συγκρίνοντάς τον με άλλους αλγορίθμους ταξινόμησης και ομαδοποίησης δεδομένων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ομαδοποίηση δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: αλγόριθμοι σμήνους, ομαδοποίηση, κατηγοριοποίηση, ταξινόμηση, εξαγωγή δεδομένων, αναγνώριση προτύπου, επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση

ABSTRACT

In this thesis we discuss the application of the “Particle Swarm Optimization Algorithm” on clustering and classification problems (supervised and unsupervised learning)

The PSO algorithm can be used to find the solution to a classification problem by transforming it to a typical optimization problem of the positions of the centers of the clusters.

Its main advantage is that it does not depend on the initialization values and performs a broader search in the search space in comparison to other algorithms. Thus, it is more difficult to be trapped in local optima.

We also present some PSO variants which are used in order to maximize its performance or to overcome particular problems.

Finally we show experimental results and we compare the PSO algorithm to other classification algorithms.

SUBJECT AREA: Data clustering and classification

KEYWORDS: clustering, classification, supervised learning, unsupervised learning, swarm intelligence, swarm algorithms, pattern recognition, data mining

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή μου, κύριο Παναγιώτη Σταματόπουλο, του οποίου η καθοδήγησή και το ενδιαφέρον ήταν τα σημαντικότερα εφόδια για να ολοκληρώσω την διπλωματική μου εργασία.

Επίσης θα ήθελα να ευχαριστήσω τον κύριο Βασίλη Ζησιμόπουλο που αξιολόγησε την διπλωματική μου εργασία.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	8
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	Error! Bookmark not defined.
1. Εισαγωγή.....	1
2. Ο ΑΛΓΟΡΙΘΜΟΣ PSO.....	2
2.1 Ανάλυση τιμών αρχικοποίησης ταχύτητας	5
3. ΟΜΑΔΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΧΡΗΣΗ ΤΟΥ PSO.....	6
3.1 Συναρτήσεις προσαρμογής.....	9
4. Προβλήματα που πρέπει να αντιμετωπίσει ο PSO	11
5. Εφαρμογές και παραλλαγές του PSO σε προβλήματα ομαδοποίησης	13
5.1 Τοπικός PSO (lbest PSO).....	13
5.2 PSO με προσομοιωμένη ανόπτηση και k-means.....	14
5.3 Combinatorial PSO	15
5.4 Ομαδοποίηση εγγράφων με χρήση PSO και k-means.....	18
5.5 k-means και Combinatorial PSO.....	20
5.6 PSO και Tabu Search υβριδικός αλγόριθμος.....	22
5.7 Immunity-clonal και PSO με σκοπό την αναγνώριση spam αλληλογραφίας	23
5.8 PSO με μαύρη τρύπα	26
5.9 Δυαδικός PSO και αναγνώριση spam μηνυμάτων	28
5.10 PSO με χρήση ρόλων στα σωματίδια για ομαδοποίηση εγγράφων Ιστού 30	
5.11 Εκπαίδευση πιθανοτικού νευρωνικού δικτύου με τη χρήση PSO	33
5.12 Ομαδοποίηση εικόνων με τη χρήση PSO	35
5.13 Μη Επιβλεπόμενη ομαδοποίηση εικόνων με δυναμική ομαδοποίηση PSO (DCPSO).....	37
6. ΥΛΟΠΟΙΗΣΗ ΜΕ ΒΑΣΗ ΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	40
6.1 Υλοποίηση μηχανισμού για φόρτωση των δεδομένων από αρχεία ARFF.....	40
6.2 Επανασχεδιασμός της κλάσης των σωματιδίων ώστε κάθε θέση του να είναι μια πολυδιάστατη υποψήφια λύση.....	41
6.3 Υλοποίηση δυναμικού καθορισμού των διαστάσεων και του αριθμού των κλάσεων ενός σωματιδίου (αναλόγως με τις διαστάσεις των δεδομένων).....	41
6.4 Υλοποίηση βοηθητικών μεθόδων για εύρεση μέγιστης και ελάχιστης τιμής κάθε διάστασης στα δεδομένα	42
6.5 Υλοποίηση διερμηνείας των μη αριθμητικών τιμών σε τιμές πραγματικών αριθμών.....	42

6.6	Υλοποίηση συνάρτησης προσαρμογής για επιβλεπόμενη και μη επιβλεπόμενη ομαδοποίηση και της μεθόδου αποτίμησής της.....	43
6.7	Υλοποίηση κλάσης ελέγχου μετά την εκπαίδευση του αλγορίθμου κατά την οποία γίνεται ουσιαστικά μέτρηση της αποτελεσματικότητάς του. 44	
6.8	Υλοποίηση 10-fold cross validation τεχνικής για την αποτίμηση των αποτελεσμάτων του αλγορίθμου.	44
6.9	Υλοποίηση μεθόδου γραμμικής μείωσης της αδράνειας όπως περιγράφεται στο [5]......	44
6.10	Υλοποίηση «σκορπίσματος ανέμου» όπως περιγράφεται στο [5].....	45
6.11	Παράμετροι εκτέλεσης του PSO ομαδοποιητή.....	45
7.	ΠΡΟΤΑΣΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ.....	46
8.	ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΩΝ.....	49
9.	ΣΥΜΠΕΡΑΣΜΑΤΑ	54
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	55
	ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	56
	ΑΝΑΦΟΡΕΣ.....	57

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Αναπαράσταση Σμήνους Σωματιδίων	2
Εικόνα 2 Αναπαράσταση σωματιδίου	8
Εικόνα 3 - Οι γειτονιές του local PSO και οι επικαλύψεις τους	13
Εικόνα 4 - Διάγραμμα ροής του TCP SO	23
Εικόνα 5 - Clonal PSO	24
Εικόνα 6 - Σύγκριση ΠΟ/Clonal/MT PSO ως προς την ταχύτητα σύγκλισης..	27
Εικόνα 7 - Σύγκριση ΠΟ/Clonal/MT PSO ως προς την εύρεση ολικού μεγίστου	27
Εικόνα 8 - Διαδικασία αναγνώρισης spam μηνυμάτων από τον δυαδικό PSO	29
Εικόνα 9 - Συγκριση βασικού PSO και PSO με ρόλους	32
Εικόνα 10 - Σύγκριση απόδοσης PNN που εκπαιδεύτηκε με PSO και BBPSO των 50 και 100 σωματιδίων.....	34
Εικόνα 11 - Αναπαράσταση των ομαδοποιημένων pixels σε μια εικόνα μαγνητικής τομογραφίας εγκεφάλου, με χρήση του PSO.....	36
Εικόνα 12 Δυναμική ανάθεση διαστάσεων και ομάδων	41
Εικόνα 13 Αναπαράσταση HashMap	42
Εικόνα 14 - Κατεύθυνση των σωματιδίων με και χωρίς βελτιστοποίηση προσωπικής παραμέτρου	47
Εικόνα 15- Σύγκριση απλής βελτίωσης με τον απλό PSO	51
Εικόνα 16- Σύγκριση βελτίωσης local με τον απλό PSO.....	52
Εικόνα 17 - Σύγκριση απόδοσης βελτιώσεων	53

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Table 4 Παράμετροι εκτέλεσης.....	45
Table 5 Σύγκριση απλού PSO με άλλους αλγόριθμους ταξινόμησης/ομαδοποίησης.....	50
Table 6 Σύγκριση PSO με βελτίωση μετάβασης και απλού PSO.....	51
Table 7 Σύγκριση PSO με τοπική βελτίωση με τον απλό PSO	52

1. Εισαγωγή

Στην εργασία αυτή θα μελετηθεί και θα αναλυθεί η χρήση του αλγόριθμου βελτιστοποίησης “Σμήνους Σωματιδίων” (Particle Swarm Optimization) πάνω σε προβλήματα ομαδοποίησης, επιβλεπόμενης και μη επιβλεπόμενης.

Ο αλγόριθμος PSO προτάθηκε αρχικά από τους Kennedy και Eberhart και βασίζεται στην αναζήτηση βέλτιστης λύσης με τη χρήση ενός “Πληθυσμού Σωματιδίων”^[5]. Όπως ένα σμήνος πουλιών ή ένα κοπάδι ψαριών κινείται στον χώρο ομαδικά για την εύρεση του καλύτερου σημείου που θα βρουν την τροφή τους έτσι και τα σωματίδια του πληθυσμού του αλγόριθμου “κινούνται”, δηλαδή παίρνουν τιμές στο χώρο αναζήτησης για να αναπαραστήσουν πιθανές λύσεις οι οποίες θα αξιολογηθούν ώστε να βρεθεί ποια είναι η βέλτιστη. Δηλαδή, κάθε θέση του σωματιδίου στον χώρο αναζήτησης αντιστοιχεί σε μια υποψήφια λύση^[1].

Σε αντίθεση με άλλους αλγόριθμους πληθυσμού, όπως πχ είναι οι γενετικοί αλγόριθμοι, ο PSO δεν έχει μεθόδους διασταύρωσης (crossover) και μετάλλαξης (mutation). Αυτό συνεπάγεται μεγαλύτερη ευκολία στην υλοποίηση, λιγότερες παραμέτρους για ρύθμιση, καθώς και γρηγορότερη σύγκλιση προς μια βέλτιστη λύση. Τα αποτελέσματα του αλγόριθμου PSO δεν εξαρτώνται από τις αρχικές θέσεις των σωματιδίων όπως γίνεται πχ με τον αλγόριθμο k-means. Ο αλγόριθμος PSO δεν πραγματοποιεί μόνο τοπική αναζήτηση καθώς λόγω της ταχύτητας και της αδράνειας, χαρακτηριστικά που θα δούμε παρακάτω, κάθε επόμενη επανάληψη συγκρατεί πληροφορία από όλα τα προηγούμενα βήματα. Έτσι η αναζήτηση γίνεται σε ευρύτερο χώρο και σε αυτό επίσης βοηθούν τα πολλά σωματίδια καθώς η αναζήτηση πραγματοποιείται παράλληλα. Αυτό συνεπάγεται επίσης περισσότερες πιθανότητες διαφυγής από τοπικά βέλτιστες λύσεις.^[3]

Για τους παραπάνω λόγους ο αλγόριθμος PSO είναι πιο αποτελεσματικός από άλλους αλγόριθμους πληθυσμού σε πολλά πρακτικά προβλήματα, όπως αυτό της ομαδοποίησης δεδομένων που θα μελετηθεί σε αυτή την εργασία.

Σχετικά με αυτό το πρόβλημα ο PSO πραγματοποιεί μια γενικευμένη αναζήτηση λύσης σε αντίθεση με τους γενετικούς αλγόριθμους ή την προσομοιωμένη απόκτηση και παρόμοιους αλγόριθμους τοπικής αναζήτησης οι οποίοι εξελίσσονται συγκρατώντας μόνο τα αποτελέσματα από την προηγούμενη επανάληψη.

Ο αλγόριθμος PSO συνήθως εφαρμόζεται με κάποιες παραλλαγές για μεγαλύτερη αποτελεσματικότητα. Σε παρακάτω κεφάλαιο θα μελετήσουμε τις σημαντικότερες παραλλαγές του σχετικά με το πρόβλημα της ομαδοποίησης δεδομένων.

2. Ο ΑΛΓΟΡΙΘΜΟΣ PSO

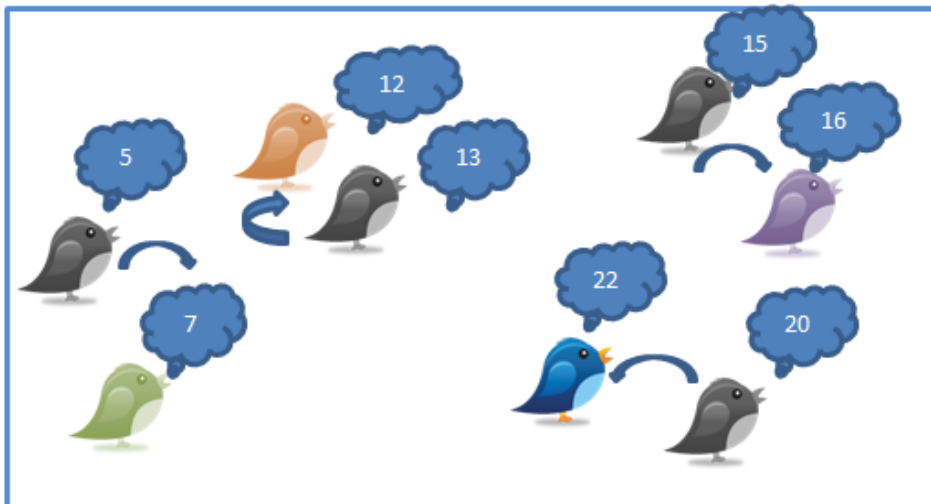
Σκοπός του αλγορίθμου είναι να βρει τη βέλτιστη λύση χρησιμοποιώντας ένα «σμήνος από σωματίδια» για την αναζήτηση μέσα στον χώρο του προβλήματος. Τα σωματίδια αυτά δεν είναι άλλο παρά πολυδιάστατα διανύσματα που αναπαριστούν θέσεις/λύσεις.

Ένα σωματίδιο λοιπόν λαμβάνει μια θέση η οποία αναπαριστά μια πιθανή λύση του προβλήματος. Η λύσεις αυτές αξιολογούνται από μια συνάρτηση προσαρμογής (fitness function) που έχει προκαθοριστεί, ώστε να προκύψει η βέλτιστη λύση.

Η διαδικασία αυτή επαναλαμβάνεται πολλές φορές ώστε να βρεθεί η βέλτιστη θέση/λύση ανάμεσα σε όλα τα σωματίδια, δηλαδή να βρεθεί το σωματίδιο εκείνο, του οποίου η θέση αναπαριστά την βέλτιστη λύση.

Σε κάθε επανάληψη κάθε σωματίδιο υπολογίζει το «προσωπικό» του βέλτιστο (την καλύτερη λύση που βρήκε μέχρι τώρα) ενώ ανάμεσα στα σωματίδια υπολογίζεται το ολικό βέλτιστο (το βέλτιστο όλων των προσωπικών βέλτιστων μέχρι τώρα).

Κατά την πρώτη επανάληψη τα σωματίδια παίρνουν τυχαίες θέσεις στον χώρο ενώ κατά τις επόμενες αλλάζουν θέση με βάση την συνάρτηση της «ταχύτητάς» τους. Η ταχύτητα είναι ένα άλλο διάνυσμα που υπολογίζεται με βάση μια συνάρτηση η οποία εμπεριέχει στοιχεία τυχαιότητας και καθορίζει τη θέση ενός σωματιδίου στην επόμενη επανάληψη. Στην πρώτη επανάληψη η ταχύτητα παίρνει μια τυχαία τιμή μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής. Η συνάρτηση αυτή λαμβάνει επίσης υπόψη τα προσωπικά και ολικά βέλτιστα με αποτέλεσμα να επηρεάζεται η θέση του σωματιδίου στην επόμενη επανάληψη και να «έλκεται» από το προσωπικό και το ολικό βέλτιστο σε διαφορετικό βαθμό.



Εικόνα 1 Αναπαράσταση Σμήνους Σωματιδίων

Έτσι λοιπόν τα 2 βασικά στοιχεία του αλγόριθμου είναι τα διανύσματα **θέση(X)** και **ταχύτητα (V)**.^[5]

Το διάνυσμα X υπολογίζεται με την παρακάτω συνάρτηση:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \vec{V}_i(t+1) \quad (1)$$

Όπου i είναι ο αριθμός του σωματιδίου του σμήνους πχ αν το σμήνος αποτελείται από N σωματίδια τότε $i=1,2,\dots,N$ και t είναι η επανάληψη, άρα εάν E ο αριθμός των συνολικών επαναλήψεων τότε $t=0,2,\dots,E-1$.

Το διάνυσμα V υπολογίζεται με την συνάρτηση:

$$\vec{V}_i(t+1) = \mathbf{W} \times \vec{V}_i(t) + \mathbf{C}_1 \times \vec{U}(0,1) \otimes (\vec{B}_i(t) - \vec{X}_i(t)) + \mathbf{C}_2 \times \vec{U}(0,1) \otimes (\vec{B}_g(t) - \vec{X}_i(t)) \quad (2)$$

Όπου τα διανύσματα \mathbf{B}_g και \mathbf{B}_i είναι το ολικό και το προσωπικό βέλτιστο αντίστοιχα, \mathbf{C}_1 ονομάζεται και εμπειρική παράμετρος και ορίζεται από τον χρήστη για να αντισταθμίζει την ταχύτητα ως προς το προσωπικό βέλτιστο. Αντίστοιχα \mathbf{C}_2 ονομάζεται και κοινωνική παράμετρος και αντισταθμίζει την ταχύτητα ως προς το ολικό βέλτιστο. $\mathbf{U}(0,1)$ είναι μια συνάρτηση που επιστρέφει ένα τυχαίο διάνυσμα που παράγεται μέσω κανονικής κατανομής στο πεδίο [0,1]. Ο τελεστής \otimes δηλώνει το εσωτερικό γινόμενο των διανυσμάτων. Τέλος ο παράγοντας \mathbf{W} είναι η «αδράνεια» του σωματιδίου και αντισταθμίζει την ταχύτητα ως προς την προηγούμενη τιμή της ταχύτητας του σωματιδίου. Συνήθως ορίζεται από τον χρήστη μια μέγιστη και μια ελάχιστη τιμή του W και η τιμή του μειώνεται γραμμικά κατά την εκτέλεση από το μέγιστο στο ελάχιστο με βάση την συνάρτηση:

$$W(t) = W_{\max} - \frac{(W_{\max} - W_{\min}) \times t}{E} \quad (3)$$

Επίσης η ταχύτητα παίρνει τιμές ανάμεσα σε ένα μέγιστο και ένα ελάχιστο όριο $[V_{\max}, V_{\min}]$.^[5]

Όπως μπορεί να διαπιστωθεί, εάν η τιμή της ταχύτητας δεν υπόκειται σε κανέναν περιορισμό, η θέση του σωματιδίου μπορεί εύκολα να βγει εκτός ορίων του χώρου αναζήτησης, να ψάχνει για λύσεις δηλαδή που δεν ορίζονται στο πρόβλημα.^{[23] [24]}

Για την εξομάλυνση του προβλήματος αυτού, χρησιμοποιείται μια τεχνική περιορισμού της ταχύτητας που ονομάζεται "velocity clamping". Για έναν χώρο αναζήτησης με τιμές στο διάστημα $[x_{\min}, x_{\max}]$ όπου $x_{\min} = -x_{\max}$, η τεχνική αυτή περιορίζει την ταχύτητα σε ένα διάστημα $[v_{\min}, v_{\max}]$ με $v_{\max} = k \times x_{\max}$ και $v_{\min} = -v_{\max}$, όπου το k είναι μια σταθερά που ορίζεται από τον χρήστη και $0,1 \leq k \leq 1$.

Σε πολλά προβλήματα ο χώρος αναζήτησης δεν έχει ως μέσο των τιμών το 0 στο διάστημα $[x_{\min}, x_{\max}]$ άρα πρέπει $x_{\min} \neq -x_{\max}$. Επομένως και ο ορισμός του διαστήματος της ταχύτητας ορίζεται ως $v_{\max} = k \times (x_{\max} - x_{\min})/2$ και $v_{\min} = -v_{\max}$.

Παρατηρώντας την εξίσωση της ταχύτητας βλέπουμε ότι αυτή επηρεάζεται από την απόσταση της τρέχουσας θέσης με το προσωπικό και το ολικό βέλτιστο ταυτόχρονα. Όσο πιο μακριά είναι η τρέχουσα θέση από τα βέλτιστα τόσο πιο πολύ μεγαλώνουν οι διαφορές $B - X$. Παράλληλα όμως οι διαφορές ρυθμίζονται από τους παράγοντες C και από την τυχαία τιμή της $U(0,1)$.

Συνεπώς θεωρούμε πως κοντά στο τοπικό και ολικό βέλτιστο βρίσκεται πιθανώς η τελική βέλτιστη λύση, έτσι οι διαφορές $B - X$ μειώνονται ώστε να μειωθεί και η ταχύτητα με σκοπό τα σωματίδια να πάρουν τις επόμενες τιμές τους γύρω από αυτές τις περιοχές.

Βλέπουμε επίσης ότι υπάρχουν παράγοντες (πχ διάνυσμα $U(0,1)$) που προσδίδουν μια τυχαιότητα στη συνάρτηση της ταχύτητας έτσι ώστε να υπάρχουν πιθανότητες διαφυγής από τοπικά βέλτιστα που μπορεί να «παγιδεύσουν» τα σωματίδια και να επηρεάσουν αρνητικά την τελική λύση.

Ψευδοκώδικας PSO

1. Για κάθε σωματίδιο i
 - a. Αρχικοποίησε τη θέση και την ταχύτητα
2. Όσο δεν φτάνουμε σε συνθήκη τερματισμού
 - a. Για κάθε σωματίδιο i
 - i. Υπολόγισε την τιμή προσαρμογής ψ_i
 - ii. Αν ψ_i καλύτερη από την τρέχουσα προσωπική βέλτιστη τότε $\psi_i =$ προσωπική βέλτιστη τιμή προσαρμογής και $p_i =$ τοπικό βέλτιστο σωματίδιο
 - iii. Αν ψ_i καλύτερη από την τρέχουσα ολική βέλτιστη τότε $\psi_i =$ ολική βέλτιστη τιμή προσαρμογής και $p_i =$ ολικό βέλτιστο σωματίδιο
 - b. Για κάθε σωματίδιο i
 - i. Ενημέρωση της ταχύτητάς του βάσει της συνάρτησής (2)
 - ii. Ενημέρωση της θέσης του βάσει της συνάρτησης (1)
 - c. Ενημέρωση της τιμής αδράνειας βάσει της εξίσωσης (3)
3. Τερματισμός

2.1 Ανάλυση τιμών αρχικοποίησης ταχύτητας

Υπάρχουν στρατηγικές υλοποίησης του PSO όσο αφορά τις επιτρεπόμενες θέσεις των σωματιδίων^{[23] [24]}

- Χωρίς περιορισμό: τα σωματίδια μπορούν να πάρουν οποιαδήποτε από τις άπειρες θέσεις στον χώρο αναζήτησης ενώ τα σωματίδια που έχουν βγει εκτός ορίων δεν λαμβάνονται πλέον υπόψη.
- Απορρόφηση: όταν ένα σωματίδιο βγει από τα επιτρεπτά όρια μιας διάστασης τότε η θέση του γίνεται ίση με το κοντινότερο (άνω ή κάτω) όριο της διάστασης αυτής ενώ η ταχύτητα του παίρνει τιμή 0.
- Τυχαία: η θέση του σωματιδίου παίρνει τυχαία τιμή εάν βγει εκτός ορίων.

Επίσης υπάρχουν 2 μέθοδοι για να αρχικοποιηθεί η ταχύτητα^{[23] [24]}

- Αρχικοποίηση βάσει κανονικής κατανομής
- Αρχική τιμή το μηδέν
- Half-Diff: Αρχική τιμή με βάση τη συνάρτηση $\frac{1}{2}(y_i - x_{i,0})$ όπου $x_{i,0}$ η αρχική τιμή του σωματιδίου και y_i μεταβλητή που παίρνει τιμές ομοιόμορφα μέσα στο πεδίο ορισμού του χώρου αναζήτησης.

Θεώρημα 1

Εάν η αρχική ταχύτητα οριστεί από την κανονική κατανομή ή τη μέθοδο half-diff τότε όλα τα σωματίδια που δεν είναι ολικό βέλτιστο ή βέλτιστο γειτονιάς θα βγουν από τα όρια.^[23]

Θεώρημα 2

Εάν η αρχική ταχύτητα οριστεί από την κανονική κατανομή κάθε σωματίδιο που είναι είτε ολικό είτε βέλτιστο γειτονιάς κατά την πρώτη επανάληψη θα βγει εκτός ορίων κατά την πρώτη επανάληψη σε, κατά μέσο όρο, $\frac{w}{4} \times n$ διαστάσεις του.^[23]

Θεώρημα 3

Εάν η ταχύτητα οριστεί με τη μέθοδο της μηδενικής αρχικής ταχύτητας τότε κάθε σωματίδιο που δεν είναι ολικό ή βέλτιστο γειτονιάς θα βγει από τον χώρο αναζήτησης στην πρώτη επανάληψη.^[23]

Συμπεραίνουμε λοιπόν ότι χρειαζόμαστε μεθόδους οριοθέτησης των θέσεων των σωματιδίων όπως για παράδειγμα την τυχαία μέθοδο ή την μέθοδο απορρόφησης που αναφέρθηκε παραπάνω.

3. ΟΜΑΔΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΧΡΗΣΗ ΤΟΥ PSO

Ο PSO χρησιμοποιείται με επιτυχία τόσο σε εφαρμογές επιβλεπόμενης όσο και σε μη επιβλεπόμενης ομαδοποίησης. (supervised and unsupervised)

Η επιβλεπόμενη ομαδοποίηση που συνήθως ονομάζεται κατηγοριοποίηση (classification) περιλαμβάνει κάποιες προκαθορισμένες κατηγορίες στις οποίες έχουν ανατεθεί δεδομένα εκπαίδευσης που κατηγοριοποιήθηκαν από κάποιον άνθρωπο. Αυτά τα δεδομένα θεωρούνται σωστά κατηγοριοποιημένα. Έπειτα οι αλγόριθμοι εκπαιδεύονται μέσω αυτής της αρχικής κατηγοριοποίησης και μπορούν πλέον να κατηγοριοποιήσουν τα επόμενα δεδομένα από μόνα τους.

Συνήθως τα ήδη κατηγοριοποιημένα δεδομένα χωρίζονται σε 2 σύνολα, εκπαίδευσης και ελέγχου (training and test data sets). Ο αλγόριθμος εκπαιδεύεται πάνω στο σύνολο εκπαίδευσης όπου και προσαρμόζεται για να μπορέσει να αναγνωρίσει πρότυπα στα οποία έπειτα μπορεί να βασιστεί για να κατηγοριοποιεί μόνος του τα δεδομένα. Μετά τη φάση εκπαίδευσης ο αλγόριθμος εφαρμόζεται πάνω στο σύνολο ελέγχου όπου και αξιολογείται η ποιότητά του.

Σε πολλές περιπτώσεις ο PSO χρησιμοποιείται μαζί με αλγόριθμους όπως τα νευρωνικά δίκτυα, k-nearest neighbor και support vector machines. Ο PSO στις περιπτώσεις αυτές χρησιμοποιείται για να εκπαιδεύσει τους άλλους αλγόριθμους, να αφαιρέσει κάποια χαρακτηριστικά από τα δεδομένα που δεν είναι χρήσιμα ή αποτελούν «θόρυβο» ή κάποιες άλλες φορές ενσωματώνεται για να δημιουργηθεί ένας υβριδικός αλγόριθμος κατηγοριοποίησης. ^{[1] [6] [9] [11] [14] [21]}

Η άλλη κατηγορία που θα μελετηθεί επίσης, είναι η μη επιβλεπόμενη ομαδοποίηση (clustering). Ο PSO μαζί με άλλους αλγόριθμους μη επιβλεπόμενης ομαδοποίησης όπως ο k-means, δημιουργούν υβριδικές μεθόδους και χρησιμοποιούνται σε προβλήματα εξαγωγής δεδομένων από έγγραφα και βάσεις δεδομένων, εικόνες καθώς και σε άλλα προβλήματα στα οικονομικά, την ιατρική και άλλους τομείς.

Κατά την μη επιβλεπόμενη ομαδοποίηση τα δεδομένα κατατάσσονται σε ομάδες, των οποίων ο αριθμός είναι άλλοτε προκαθορισμένος από τον χρήστη και άλλοτε προσδιορίζεται και αυτός από τον αλγόριθμο. Οι ομάδες δημιουργούνται με βάση κάποια αντικειμενική συνάρτηση απόστασης (πχ ευκλείδεια απόσταση) των σημείων από τα σημεία-κέντρα των ομάδων. Σκοπός των αλγόριθμων μη επιβλεπόμενης ομαδοποίησης είναι να μεγιστοποιήσουν την ομοιότητα μεταξύ των δεδομένων που ανήκουν στην ίδια ομάδα και παράλληλα να ελαχιστοποιήσουν την ομοιότητα μεταξύ των στοιχείων που ανήκουν σε διαφορετικές ομάδες.

Όπως αναφέρθηκε παραπάνω υπάρχουν 2 κατηγορίες αλγόριθμων μη επιβλεπόμενης ομαδοποίησης που θα συναντήσουμε και παρακάτω στις

πρακτικές εφαρμογές, οι επιμεριστικοί και οι ιεραρχικοί (partitional and hierarchical). Οι επιμεριστικοί διαχωρίζουν τα δεδομένα σε έναν αριθμό μη επικαλυπτόμενων ομάδων που έχει καθοριστεί από τον χρήστη εκ των προτέρων ενώ οι ιεραρχικοί δημιουργούν ένα δέντρο με όλες τις πιθανές ομαδοποιήσεις που μπορούν να προκύψουν. Η δημιουργία του δέντρου ξεκινάει είτε με μια μεγάλη ομάδα η οποία συνεχώς διαιρείται σε μικρότερες ώστε να προκύψουν οι άλλες ομαδοποιήσεις, είτε ξεκινάει από 1 ομάδα για κάθε στοιχείο οι οποίες ενώνονται ώστε να δημιουργηθούν οι υπόλοιπες ομαδοποιήσεις (top-down και bottom-up αντίστοιχα). Οι ιεραρχικοί αλγόριθμοι θεωρούνται ποιοτικότεροι στα αποτελέσματά τους αλλά έχουν αρκετά μεγαλύτερη πολυπλοκότητα. ^{[5] [7] [8] [10] [16] [17] [18] [19]}

Η χρήση του PSO έχει αποδειχθεί τόσο αποτελεσματική όσο και γρήγορη στην επίλυση προβλημάτων ομαδοποίησης. Ο PSO μπορεί να αντιμετωπίσει το πρόβλημα της ομαδοποίησης ως ένα πρόβλημα βελτιστοποίησης της ομοιότητας των δεδομένων και τις διαφορετικότητας των ομάδων μέσω της εύρεσης των βέλτιστων κέντρων των ομάδων και έτσι να εφαρμοστεί και σε αυτόν τον χώρο.

Σε κάθε επανάληψη τα σωματίδια του PSO αναπαριστούν τα πιθανά κέντρα των ομάδων. Εάν N οι ομάδες και C οι διαστάσεις κάθε μίας, τότε ένα διάνυσμα θέσης σωματιδίου έχει $N \times C$ διαστάσεις. Έτσι σε κάθε βήμα αξιολογούμε παράλληλα K πιθανές ομαδοποιήσεις, όπου K ο αριθμός των σωματιδίων. Η αξιολόγηση γίνεται με τη χρήση συνάρτησης προσαρμογής. ^[5]

$$\mathbf{X}_i = \{m_1, m_2, \dots, m_c\}$$
$$m_j = \{x_{j1}, x_{j2}, \dots, x_{jd}\}$$

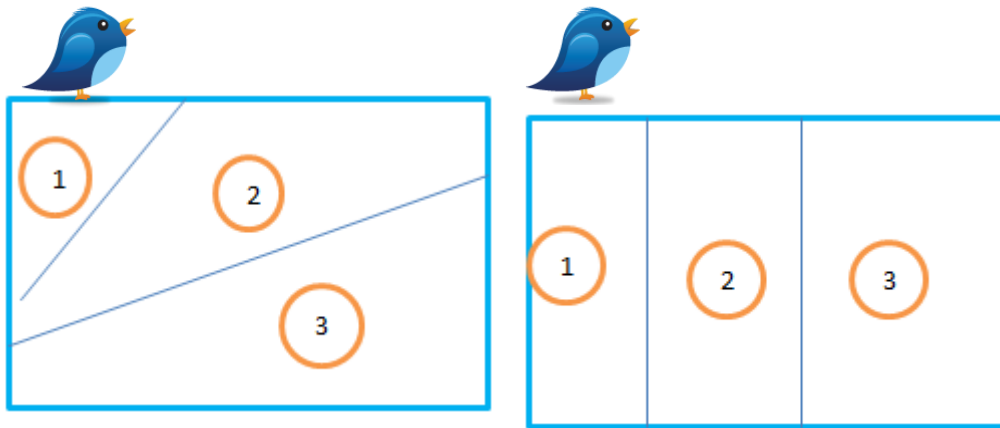
X_i : διάνυσμα θέσης σωματιδίου, m_j : διάνυσμα που αναπαριστά το κέντρο της j κλάσης που αναπαρίσταται από το σωματίδιο i .

Για παράδειγμα στην επιβλεπόμενη ομαδοποίηση η συνάρτηση μπορεί να είναι η διαφορά της αναπαράστασης του κέντρου του σωματιδίου σε σχέση με την κατηγοριοποίηση των δεδομένων του συνόλου εκπαίδευσης. Ενώ στη μη επιβλεπόμενη μια συνάρτηση συσχέτισης των αποστάσεων των δεδομένων εντός της ίδιας ομάδας προς την απόσταση μεταξύ των ομάδων. Και στις 2 περιπτώσεις η βέλτιστη ομαδοποίηση είναι αυτή που ελαχιστοποιεί την τιμή των συναρτήσεων προσαρμογής. ^{[5][4]}

Η διαφορά του PSO σε σχέση με άλλους αλγόριθμους ομαδοποίησης, όπως για παράδειγμα τον k -means, βρίσκεται στο ότι πραγματοποιεί μια πιο γενική αναζήτηση βέλτιστου αντίθετα με άλλους αλγόριθμους που περιορίζονται στην

τοπική αναζήτηση. Για παράδειγμα ο k-means που αναφέρθηκε προηγουμένως ξεκινάει με μια τυχαία επιλογή κέντρων τα οποία βελτιστοποιεί σε κάθε επανάληψη. Αυτό σημαίνει ότι ο αλγόριθμος εξελίσσεται με βάση το αποτέλεσμα της προηγούμενης του επανάληψης και μόνο. Αυτό περιορίζει τον αλγόριθμο στο να πραγματοποιεί την αναζήτηση σε ένα πολύ μικρό χώρο γύρω από τα αρχικώς επιλεγμένα κέντρα.

Αντίθετα, ο PSO διαχειρίζεται διαφορετικά την εύρεση των βέλτιστων κέντρων των ομάδων. Ξεκινάει αρχικά μια γενική αναζήτηση σε έναν ευρύ χώρο ενώ σε δεύτερη φάση περιορίζεται σε μια τοπική αναζήτηση γύρω από τα βέλτιστα που έχουν βρεθεί από την γενική αναζήτηση.



Εικόνα 2 Αναπαράσταση σωματιδίου

Αυτή η διαφορά από τη γενική φάση στην τοπική είναι εφικτή με τη χρήση της ταχύτητας που είναι μεγάλη όταν βρισκόμαστε μακριά από τα βέλτιστα, άρα ένα σωματίδιο με μεγάλη ταχύτητα παίρνει τιμές που απέχουν περισσότερο μεταξύ τους και έτσι δεν περιοριζόμαστε σε τοπικές τιμές γύρω από την αρχική θέση. Όταν όμως το σωματίδιο βρεθεί κοντά σε βέλτιστο τότε η ταχύτητα μειώνεται και το σωματίδιο παίρνει τιμές τοπικά γύρω από το βέλτιστο.

Η χρήση πολλών σωματιδίων, δηλαδή ο έλεγχος πολλών πιθανών λύσεων σε κάθε επανάληψη, μπορεί να καλύψει ευρύτερο χώρο στην αναζήτηση. Οι αρχικές επαναλήψεις θεωρούνται ως το τμήμα της γενικής αναζήτησης του PSO το οποίο χαρακτηρίζεται από την μεγάλη ταχύτητα που έχουν τα σωματίδια. Στη συνέχεια σταδιακά μειώνεται η ταχύτητα άρα και ο χώρος αναζήτησης και τα σωματίδια παίρνουν τιμές γύρω από τα τρέχοντα βέλτιστα. Δηλαδή έχουμε μια μετακίνηση από το γενικό στάδιο αναζήτησης στο τοπικό.

Ο χρόνος μετακίνησης από το ένα στάδιο στο άλλο καθορίζεται με βάση τις επιλογές μας στις τιμές των παραμέτρων των παραπάνω συναρτήσεων.

Συμπεραίνουμε λοιπόν πως η βέλτιστη λύση (στην περίπτωση μας η βέλτιστη ομαδοποίηση) μπορεί να βρεθεί αποτελεσματικότερα εάν ο χρόνος του σταδίου της γενικής αναζήτησης του PSO είναι αρκετά μεγάλος, διότι ο

αλγόριθμος θα αναζητήσει βέλτιστες λύσεις σε όλο το εύρος του χώρου αναζήτησης και δεν θα παγιδευτεί σε τοπικά βέλτιστες λύσεις.

3.1 Συναρτήσεις προσαρμογής

Όπως είδαμε προηγουμένως έχουμε 2 μεθόδους ομαδοποίησης, την επιβλεπόμενη και την μη επιβλεπόμενη που ομαδοποιούν με διαφορετική φιλοσοφία τα δεδομένα. Συνεπώς για κάθε μέθοδο έχουμε διαφορετική συνάρτηση προσαρμογής.

Σκοπός της επιβλεπόμενης ομαδοποίησης είναι η εκπαίδευση βάσει ενός συνόλου δεδομένων που ήδη έχουν ομαδοποιηθεί από άνθρωπο. Έτσι η συνάρτηση προσαρμογής για την συγκεκριμένη μέθοδο πρέπει να αποτιμά την απόσταση που έχει κάθε δείγμα από τα δεδομένα από την αντίστοιχη κλάση που αντιπροσωπεύεται μέσα σε κάθε σωματίδιο. Αν για παράδειγμα το πρώτο δείγμα ανήκει στην 2^η κλάση τότε πρέπει να συγκριθεί η θέση του με τη θέση της 2^{ης} κλάσης του κάθε σωματιδίου. Όσο μικρότερη η απόσταση τόσο πιο κοντά σε βέλτιστη λύση βρίσκεται το σωματίδιο.

Τύπος συνάρτησης προσαρμογής:^[5]

$$\Psi_c(t) = \frac{1}{DTrain} \times \sum_{i=1}^{DTrain} d(y_i^c - x_i^c)$$

Όπου c η κλάση που αντιστοιχεί στο τρέχον δείγμα, y_i το διάνυσμα θέσης του δείγματος, x_i το διάνυσμα θέσης του κέντρου της κλάσης c όπως αναπαρίσταται από το σωματίδιο i και $d(\dots)$ η απόστασή τους. DTrain είναι ο αριθμός των δειγμάτων από τα οποία αποτελείται το σύνολο εκπαίδευσης.

Σκοπός της μη επιβλεπόμενης ομαδοποίησης είναι να ομαδοποιήσει τα δεδομένα έτσι ώστε τα δεδομένα μιας ομάδας να έχουν μεταξύ τους τη μικρότερη δυνατή απόσταση (Intra distance) ενώ τα κέντρα των κλάσεων μεταξύ τους τη μεγαλύτερη δυνατή απόσταση (Inter distance). Έτσι θα προκύψει η βέλτιστη ομαδοποίηση.

Τύπος συνάρτησης προσαρμογής:^[4]

$$\Psi(t) = \frac{\sum_{k=1}^N \sum_{x_i \in C_k} (x_i - m_k)^2 / C_k}{\sum_{i,j \in C} (m_i - m_j)^2}$$

Όπου N ο αριθμός των κλάσεων, m_k το κέντρο της κλάσης k, C_k ο αριθμός των δειγμάτων στην κλάση k, x_i δείγμα που ανήκει στην τρέχουσα κλάση k, ενώ i, j με $1 \leq i \neq j \leq$ είναι 2 διαφορετικές κλάσεις και m τα κέντρα τους.

Δηλαδή η συνάρτηση προσαρμογής υπολογίζει το άθροισμα σε όλες τις κλάσεις του μέσου όρου των αποστάσεων των δειγμάτων από το κέντρο της κάθε κλάσης που ανήκουν ως προς το άθροισμα της απόστασης όλων των κέντρων των κλάσεων μεταξύ τους. Το τελευταίο άθροισμα (Inter distance)

πρέπει να είναι μεγιστοποιείται ώστε η τιμή του κλάσματος να μικραίνει ενώ το πρώτο άθροισμα(Intra distance) πρέπει να ελαχιστοποιείται.

4. Προβλήματα που πρέπει να αντιμετωπίσει ο PSO

Η ομαδοποίηση με τον PSO φαίνεται να επηρεάζει σημαντικά από 2 κυρίως παράγοντες^{[5][14]}:

- Μεγάλος αριθμός διαστάσεων των δεδομένων
- Δεδομένα διαφορετικών τύπων (πραγματικοί αριθμοί, συμβολικές τιμές)
- Τοπικά βέλτιστα
- Χρόνος Εκτέλεσης
- Τοπική Αναζήτησης

Αναφορικά με το πρώτο πρόβλημα παρατηρείται μείωση της απόδοσης και της ακρίβειας του PSO όταν τα δεδομένα περιλαμβάνουν πολλούς ασήμαντους όρους σε συνδυασμό με τις αυξημένες διαστάσεις των διανυσμάτων αναπαράστασης των λύσεων (πχ πολλές λέξεις σε έγγραφα, που δεν δίνουν νόημα.) που θεωρείται «θόρυβος» στα δεδομένα. Επίσης ο αλγόριθμος επηρεάζεται αρνητικά όταν ο αριθμός των κλάσεων ομαδοποίησης των δεδομένων είναι μεγάλος.

Αναφορικά με το δεύτερο πρόβλημα παρουσιάζεται δυσκολία αναπαράστασης της πληροφορίας με αποτελεσματικό τρόπο, όταν τα δεδομένα είναι ετερογενή και παίρνουν τιμές διαφορετικού τύπου. Για παράδειγμα μπορούμε να έχουμε αριθμητικές τιμές, συνεχείς αλλά και διακριτές, όπως επίσης και ονομαστικές τιμές στα δεδομένα μας.

Αν λάβουμε υπόψη μας μια διαδικασία τράπεζας που πρέπει να κατηγοριοποιήσει κάποιες αιτήσεις για χορήγηση δανείων ως ακίνδυνες ή ρισκοκίνδυνες τότε μπορούμε να φανταστούμε ότι κάποια δεδομένα θα είναι διακριτά αριθμητικά (ηλικία) κάποια συνεχή αριθμητικά (μισθός) και κάποια ονομαστικά (οικογενειακή κατάσταση).

Τα δεδομένα αυτά περνούν από ένα στάδιο επεξεργασίας και μετασχηματισμού σε αριθμητικές τιμές. Το πρόβλημα όμως είναι πώς αυτή η νέα αναπαράσταση θα διατηρήσει τον βαθμό της σημασίας και την σημασία των σχέσεων της με τις άλλες μεταβλητές; Πώς αυτή η αναπαράσταση μπορεί να μας δώσει την επιθυμητή πληροφορία;

Για τους παραπάνω λόγους έχουν υλοποιηθεί διάφορες παραλλαγές του βασικού PSO ώστε να αντιμετωπιστούν τα προβλήματα.

Αποτελεσματικές Λύσεις

Για το πρόβλημα των πολλών διαστάσεων προτείνονται οι λύσεις του ελέγχου θέσης των σωματιδίων και η εισαγωγή μιας χαοτικής συμπεριφοράς στα σωματίδια.

Η νέα θέση ενός σωματιδίου μπορεί να περιοριστεί σε ένα διάστημα με την συνάρτηση^[5]:

$x_{i,k}(t+1) = \text{MIN}(\text{MAX}(x_{i,k}(t) + u_{i,k}(t+1), x_{\min}), x_{\max})$, δηλαδή ο έλεγχος και ο περιορισμός κάθε νέας τιμής θέσης ανάμεσα στο x_{\min} και x_{\max} .

όπου k η διάσταση και x_i το σωματίδιο.

Επίσης η χαοτική συμπεριφορά μπορεί να εφαρμοστεί με την συνάρτηση «ανέμου»^[5]:

$vw(t+1) = vw(t) + u_{op} \times \text{rand}() + u_{su} \times \text{rand}()$, τυχαία τιμή που διαφοροποιεί την νέα ταχύτητα του σωματιδίου με χαοτικό τρόπο για αποφυγή τοπικών βέλτιστων.

όπου $u_{op} = -1$ και $u_{su} = 1$ και αναπαριστούν τη φορά του ανέμου που άλλοτε είναι ίδια με το διάνυσμα του σωματιδίου και άλλοτε αντίθετη.

Ο «άνεμος» προστίθεται στη συνάρτηση της θέσης του PSO και μπορεί να χρησιμοποιηθεί μαζί με τον έλεγχο θέσης, συνδυασμός που είναι αρκετά πιο αποτελεσματικός.

Για το πρόβλημα της μετατροπής των τιμών από συνεχείς σε δυαδικές έχουμε την παραλλαγή του διακριτού PSO την οποία θα συναντήσουμε και στο κεφάλαιο 5 με τις εφαρμογές σε θέματα ομαδοποίησης^[5].

Συνοπτικά, η συνάρτηση θέσεις γίνεται:

$$x_i^{t+1} = \begin{cases} 1, & \text{if } \text{rand}_i^{t+1} < \text{sig}(V_i^{t+1}) \\ 0, & \text{otherwise} \end{cases}$$

όπου $\text{sig}()$ είναι η σιγμοειδής συνάρτηση. Η μετατροπή γίνεται με τυχαίο τρόπο και οι τιμές της θέσης στην επόμενη επανάληψη εξαρτώνται από το εάν μια τυχαία τιμή είναι μικρότερη ή μεγαλύτερη από την τιμή της σιγμοειδούς συναρτήσεως που έχει ως παράμετρο την τιμή της ταχύτητας στην επόμενη επανάληψη.

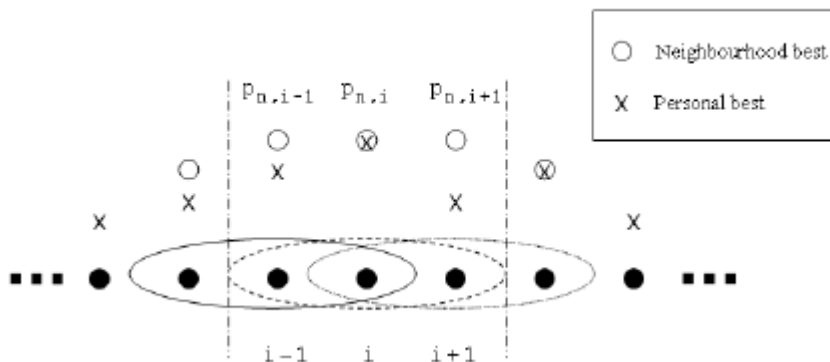
5. Εφαρμογές και παραλλαγές του PSO σε προβλήματα ομαδοποίησης

5.1 Τοπικός PSO (lbest PSO)

Μια βασική παραλλαγή του PSO είναι ο διαχωρισμός του σμήνους σε γειτονιές^[14]. Για παράδειγμα μπορεί να ακολουθηθεί το πρότυπο του δακτυλίου κι έτσι κάθε σωματίδιο να συγκρίνει τη θέση του με το επόμενο του και το προηγούμενό του αποκλειστικά. Θεωρώντας τα σωματίδια σε σχήμα δακτυλίου και ότι το μέγεθος γειτονιάς είναι 3, το σωματίδιο i θα δημιουργήσει γειτονιά με το $i-1$ και το $i+1$. Τα 3 σωματίδια και κάθε άλλη τριάδα σωματιδίων θα συγκρίνουν τις θέσεις τους και θα εξάγεται η βέλτιστη, μεταξύ της κάθε τριάδας.

Έτσι το ολικό βέλτιστο αντικαθίσταται από το βέλτιστο της γειτονιάς με σκοπό τα σωματίδια να μην επηρεάζονται από τη θέση μίας μόνο κοινής βέλτιστης λύσης αλλά από πολλές τοπικά βέλτιστες.

Είναι προφανές ότι αν θέσουμε το μέγεθος της γειτονιάς ίσο με το μέγεθος του σμήνους, τότε προκύπτει ο global PSO.



Εικόνα 3 - Οι γειτονιές του local PSO και οι επικαλύψεις τους.

Παρατηρούμε ότι η θέση του σωματιδίου i δεν είναι μόνο η προσωπική του βέλτιστη αλλά και η βέλτιστη στη γειτονιά. Έτσι το ολικό βέλτιστο είναι η θέση του i και για τα διπλανά του σωματίδια. Τα σωματίδια έξω από τη γειτονιά δεν επηρεάζονται από αυτό το βέλτιστο.

Ο αλγόριθμος local PSO

1. Αρχικοποίηση του σμήνους.
2. Για κάθε σωματίδιο i
 - i. Εάν η τρέχουσα θέση είναι καλύτερη από την προσωπική βέλτιστη όρισέ την σαν νέα προσωπική βέλτιστη.
3. Για κάθε σωματίδιο i
 - i. Εύρεση του βέλτιστου της γειτονιάς
4. Για κάθε σωματίδιο i
 - i. Ανανέωση της θέσης και της ταχύτητας των σωματιδίων με χρήση των συναρτήσεων του PSO. Η μόνη διαφορά είναι στην συνάρτηση της ταχύτητας όπου στη θέση του διανύσματος του ολικού βέλτιστου θα μπαίνει το διάνυσμα του αντίστοιχου προς το σωματίδιο i , βέλτιστου της γειτονιάς.

5.2 PSO με προσομοιωμένη ανόπτηση και k-means

Ο συγκεκριμένος υβριδικός αλγόριθμος συνδυάζει τα πλεονεκτήματα του PSO στην γενική αναζήτηση βέλτιστου, της προσομοιωμένης ανόπτησης στην τοπική αναζήτηση και του k-means στην αποτελεσματική ομαδοποίηση^{[16][17]}.

Περιγραφή του αλγόριθμου

1. Αρχικοποίηση του σμήνους.
2. Υπολογισμός συνάρτησης προσαρμογής του σμήνους.
3. Ταξινόμηση των σωματιδίων με βάση το αποτέλεσμα της συνάρτησης προσαρμογής.
4. Επιλογή ολικού βέλτιστου σωματιδίου.
5. Εκτέλεση προσομοιωμένης ανόπτησης με αρχική τιμή το τρέχον ολικό βέλτιστο.
6. Αν το βέλτιστο της ανόπτησης είναι καλύτερο από το τρέχον τότε αυτό θα γίνει το νέο τρέχον βέλτιστο.
7. Για κάθε σωματίδιο i :
 - a. Υπολογισμός του προσωπικού του βέλτιστου.
 - b. Υπολογισμός ταχύτητας και ανανέωση θέσης του i .
8. Υπολογισμός συνάρτησης προσαρμογής του σμήνους.
9. Ταξινόμηση των σωματιδίων με βάση το αποτέλεσμα της συνάρτησης προσαρμογής.
10. Επανάληψη από το βήμα 4 μέχρι να ικανοποιηθούν οι συνθήκες τερματισμού.
11. Εκτέλεση του k-means έχοντας σαν αρχικά κέντρα κλάσεων τα κέντρα που αναπαριστά το τρέχον ολικό βέλτιστο.

12. Αν το αποτέλεσμα του k-means είναι καλύτερο από το τρέχον βέλτιστο τότε θεωρείται ως η τελική λύση. Αλλιώς η τελική λύση είναι το τρέχον βέλτιστο.
13. Επανάληψη μέχρι να ικανοποιηθούν οι συνθήκες τερματισμού.

Παρόλο που ο PSO είναι αποτελεσματικός στη εύρεση βέλτιστης λύσης η δυνατότητα τοπικής αναζήτησης γύρω από το βέλτιστο είναι μειωμένη λόγω της αδράνειας w που θέτει την ταχύτητα όλο και πιο κοντά στο 0. Το βέλτιστο αυτό είναι πολύ πιθανό να είναι όμως τοπικό με αποτέλεσμα το σωματίδιο αυτό να παρασύρει τα υπόλοιπα σωματίδια στη λύση αυτή. Έτσι χρησιμοποιείται ο αλγόριθμος προσομοιωμένης ανόπτωσης ο οποίος είναι από τους πιο αποτελεσματικούς στην τοπική αναζήτηση. Παίρνοντας ως παράμετρο το βέλτιστο του PSO γίνεται ακόμα πιο αποδοτικός αφού παρακάμπτεται το πρόβλημα της ανόπτωσης όσο αφορά το σημείο εκκίνησης το οποίο μπορεί να μην είναι το βέλτιστο για τη σωστή σύγκλιση του αλγόριθμου προς ένα μη τοπικό βέλτιστο.

Τέλος εφαρμόζεται ο k-means που παρουσιάζει παρόμοιο πρόβλημα με τον αλγόριθμο της ανόπτωσης, καθώς μπορεί να πέσει σε τοπικά βέλτιστες λύσεις εάν τα κέντρα των κλάσεων κατά την αρχικοποίηση δεν είναι κατάλληλα, κάτι πολύ πιθανό να συμβεί εφόσον επιλέγονται τυχαία. Με την χρήση ως αρχικών κέντρων, τα σημεία που υποδεικνύει το ολικό βέλτιστο του PSO, το πρόβλημα αυτό παρακάμπτεται επιτυχώς.

Ο k-means επιλέγεται σε αυτή τη φάση γιατί απαιτεί λιγότερες επαναλήψεις από τον PSO για να καταλήξει στη βέλτιστη λύση ομαδοποίησης.

Έτσι η υβριδική μορφή προτιμάται αφού εξομαλύνει τα μειονεκτήματα των επιμέρους αλγόριθμων που θα υπήρχαν εάν αυτοί ενεργούσαν ξεχωριστά και παράλληλα εκμεταλλεύεται αποτελεσματικά τα πλεονεκτήματα τους στα διάφορα στάδια τη διαδικασίας ομαδοποίησης που είναι η αναζήτηση βέλτιστων κέντρων των κλάσεων και ο υπολογισμός των κλάσεων.

5.3 Combinatorial PSO

Το διάνυσμα $Y_i^t = \{y_{i1}^t, y_{i2}^t, \dots, y_{in}^t\}$ συνδέεται με το διάνυσμα των σωματιδίων

$$X_i^t = \{x_{i1}^t, x_{i2}^t, \dots, x_{in}^t\}.^{[6]}$$

Κάθε διάσταση του Y παίρνει τιμές μεταξύ $\{-1,0,1\}$ σύμφωνα με την τιμή της αντίστοιχης διάστασης του διανύσματος X . Στην πραγματικότητα το Y είναι μια προσωρινή μεταβλητή που χρησιμεύει στην μεταφορά από την συνδυαστική κατάσταση στην συνεχή.

Η τιμή της μεταβλητής Y ορίζεται από τον τύπο:

$$y_{ij}^t = \begin{cases} 1, & \text{if } x_{ij}^t = G_j^t \\ -1, & \text{if } x_{ij}^t = p_{ij}^t \\ 1 \text{ or } -1 \text{ randomly,} & \text{if } x_{ij}^t = G_j^t = p_{ij}^t \\ 0, & \text{otherwise} \end{cases}$$

όπου G_j^t το ολικό βέλτιστο και p_{ij}^t το προσωπικό βέλτιστο.

Η συνάρτηση της ταχύτητας στην περίπτωση αυτή είναι:

$$v_{ij}^t = w \times v_{ij}^{t-1} + r_1 \times c_1 \times d_1 + r_2 \times c_2 \times d_2$$

όπου $d_1 = -1 - y_{ij}^{t-1}$ είναι η απόσταση μεταξύ της τρέχουσας λύσης x_{ij}^{t-1} και του βέλτιστου του i -οστού στοιχείου.

και $d_2 = 1 - y_{ij}^{t-1}$ είναι η απόσταση μεταξύ της τρέχουσας λύσης x_{ij}^{t-1} και του ολικού βέλτιστου του σμήνους.

Με βάση την συνάρτηση αυτή η ταχύτητα V εξαρτάται από το διάνυσμα Y :

- Αν $x_{ij}^t = G_j^t$ τότε $y_{ij}^t = 1$ άρα το $d_2 = 0$ και το $d_1 = -2$ οπότε και η ταχύτητα μειώνεται.
- Αν $x_{ij}^t = p_{ij}^t$ τότε $y_{ij}^t = -1$ άρα το $d_2 = 2$ και το $d_1 = 0$ οπότε και η ταχύτητα αυξάνεται.
- Αν $x_{ij}^t \neq G_j^t$ και $x_{ij}^t \neq p_{ij}^t$ τότε $y_{ij}^t = 0$ άρα το $d_2 = 1$ και το $d_1 = -1$ οπότε και η ταχύτητα εξαρτάται μόνο από τις παραμέτρους r_1, c_1, r_2, c_2 .
- Τέλος αν $x_{ij}^t = G_j^t = p_{ij}^t$ τότε y_{ij}^t παίρνει τυχαία μια τιμή εκ των $\{-1, 1\}$ άρα η ταχύτητα παίρνει κατεύθυνση αντίθετη από αυτή του διανύσματος y_{ij}^t .

Χρησιμοποιώντας την παρακάτω μεθοδολογία θα παραχθεί μια νέα λύση για την επόμενη επανάληψη του αλγορίθμου:

$$\lambda_{ij}^t = y_{ij}^{t-1} + v_{ij}^t$$

όπου λ είναι μια βοηθητική ενδιάμεση μεταβλητή, στη συνέχεια η μεταβλητή y_{ij}^t παίρνει τιμή ως εξής:

$$y_{ij}^t = \begin{cases} 1, & \text{if } \lambda_{ij}^t > a \\ -1, & \text{if } \lambda_{ij}^t < -a \\ 0, & \text{otherwise} \end{cases}$$

Εάν επιλέξουμε μικρή παράμετρο α τότε ωθούμε την τελική λύση να πάρει την τιμή του ολικού ή του προσωπικού βέλτιστου όπως θα δούμε αμέσως παρακάτω. Σε αντίθετη περίπτωση, για μεγάλο α , η λύση θα πάρει μια τυχαία τιμή ώστε να υπάρχει η απαιτούμενη διαφοροποίηση για αποφυγή τοπικών βέλτιστων.

Τέλος η νέα λύση παράγεται με την παρακάτω συνάρτηση:

$$x_{ij}^t = \begin{cases} G_j^{t-1}, & \text{if } y_{ij}^t = 1 \\ p_{ij}^{t-1}, & \text{if } y_{ij}^t = -1 \\ \text{a random number,} & \text{otherwise} \end{cases}$$

Παρακάτω βλέπουμε ένα παράδειγμα με τη μεθοδολογία για την παραγωγή μιας νέας λύσης:

Τιμές για ολικό, τοπικό βέλτιστο και τρέχον σωματίδιο

P_i^{t-1}	2	1	3	2	3	1
G_i^{t-1}	3	2	3	1	2	2
X_i^{t-1}	1	1	1	2	3	2

Βήμα 1

Y_i^{t-1}	0	1	0	-1	-1	1
-------------	---	---	---	----	----	---

Βήμα 1

V_i^t	-1.2	-0.9	0.5	0.7	0.9	0.3
---------	------	------	-----	-----	-----	-----

Βήμα 3

λ_i^t	-1.2	0.1	0.5	-0.3	-0.1	1.3
---------------	------	-----	-----	------	------	-----

Βήμα 2

Y_i^t	-1	0	1	0	0	1
---------	----	---	---	---	---	---

Βήμα 3

X_i^t	2	3	3	1	1	2
---------	---	---	---	---	---	---

5.4 Ομαδοποίηση εγγράφων με χρήση PSO και k-means

Μια από τις βασικότερες εφαρμογές του αλγόριθμου PSO είναι στην ομαδοποίηση εγγράφων. Σκοπός της ομαδοποίησης των εγγράφων είναι η εύρεση δομών και ομοιοτήτων στα δεδομένα του εγγράφου με σκοπό αυτά να ομαδοποιηθούν κατάλληλα. Έτσι μπορούμε να επιτύχουμε την αυτόματη εξαγωγή χρήσιμων πληροφοριών όπως είναι η εύρεση των θεματικών εννοιών στις οποίες αναφέρεται το έγγραφο άρα η αυτόματη εξαγωγή της θεματολογίας του. Επομένως θα μπορούσαμε για παράδειγμα, ανάμεσα σε ένα σύνολο εγγράφων να ξεχωρίσουμε ποιά αναφέρονται σε μαθηματικά, ποιά σε πληροφορική, ποιά σε ειδήσεις κλπ.

Πριν δούμε τη χρήση του PSO πάνω σε αυτό το αντικείμενο ας δούμε πρώτα κάποιες βασικές πληροφορίες για το πώς αναπαριστούμε ένα έγγραφο.

Ένα έγγραφο μπορεί να αναπαρασταθεί με ένα διάνυσμα χαρακτηριστικών. Για παράδειγμα με τη χρήση του Vector Space Model ένα έγγραφο αναπαρίσταται από το διάνυσμα $d = \{w_1, w_2, \dots, w_n\}$ όπου w_i είναι το βάρος ενός όρου, δηλαδή μιας λέξης, t_i .^{[1][7]}

Ένας βασικός τρόπος υπολογισμού του βάρους είναι ο αλγόριθμος TF-IDF.

Το βάρος για τον όρο i σε ένα έγγραφο j υπολογίζεται ως εξής:

$$W_{ji} = tf_{ji} \times idf_i = tf_{ji} \times \log_2 (n/df_i)$$

όπου

tf_{ji} είναι η επαναλήψεις του όρου i στο έγγραφο j , df_i είναι η συχνότητα του όρου μέσα σε όλα τα έγγραφα και n είναι το σύνολο των εγγράφων.

Το μέτρο ομοιότητας μεταξύ 2 εγγράφων είναι η ευκλείδεια απόσταση:

$$d(m_p, m_j) = \sqrt{\sum_{k=1}^{d_m} (m_{pk} - m_{jk})^2 / d_m}$$

όπου m_p , m_j είναι τα διανύσματα των 2 εγγράφων, d_m είναι ο αριθμός των διαστάσεων και k είναι η τρέχουσα διάσταση.

Άλλο μέτρο ομοιότητας είναι το cosine correlation:

$$\text{Cos}(m_p, m_j) = \frac{m_p^t m_j}{|m_p| |m_j|}$$

Συνοπτικά ο αλγόριθμος k-means ορίζεται ως εξής:

- Επιλογή τυχαίων k σημείων ως κέντρων για την αρχικοποίηση των k κλάσεων
- Ανάθεση κάθε διανύσματος εγγράφου στην κοντινότερη κλάση
- Επανακαθορισμός των κέντρων με τη χρήση της συνάρτησης:
 - $c_j = \frac{1}{n_j} \sum_{v \in S_j} d_j$
- επανάληψη από το δεύτερο βήμα μέχρι να επιτευχθεί σύγκλιση

Το μειονέκτημα του k-means είναι ότι εξαρτάται πολύ από την αρχική τυχαία επιλογή των κέντρων. Παρακάτω θα δούμε μια υβριδική μέθοδο όπου αρχικά χρησιμοποιείται ο PSO ακριβώς για να λύσει αυτό το πρόβλημα.

Όπως αναφέραμε στην εισαγωγή, το πρόβλημα της ομαδοποίησης μπορεί να αναχθεί στην βελτιστοποίηση των κέντρων άρα μπορεί και να εφαρμοστεί ο PSO.

Κάθε σωματίδιο ορίζεται ως ένα σύνολο διανυσμάτων τα οποία είναι τα κέντρα.

$$X_i = (C_1, C_2, \dots, C_i, \dots, C_k)$$

Ο αλγόριθμος ορίζεται ως εξής:

- Επιλογή τυχαίων διανυσμάτων εγγράφων ως κέντρων, από κάθε σωματίδιο
- Για κάθε σωματίδιο
 - Ανάθεση κάθε διανύσματος στο κοντινότερο κέντρο
 - Υπολογισμός συνάρτησης προσαρμογής
 - Υπολογισμός επόμενης λύσης με τη χρήση των συναρτήσεων θέσης και ταχύτητας του PSO.
- Τερματισμός αν φτάσουμε τις μέγιστες επαναλήψεις ή αν δεν υπάρχει σημαντική αλλαγή στα κέντρα.

Ως συνάρτηση προσαρμογής μπορούμε να χρησιμοποιήσουμε την συνάρτηση:

$$f = \frac{\sum_{i=1}^{N_c} \left\{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \right\}}{N_c}$$

όπου m_{ij} είναι το j -οστό έγγραφο που ανήκει στην κλάση i , O_i είναι το κέντρο της κλάσης i , $d(o,m)$ είναι η απόσταση του εγγράφου από το κέντρο της κλάσης, P_i είναι ο αριθμός των εγγράφων της κλάσης i και N_c είναι ο αριθμός των κλάσεων.

Παρόλο που ο αλγόριθμος PSO μπορεί να είναι αρκετά πιο αποτελεσματικός στην εύρεση της βέλτιστης ομαδοποίησης είναι εξαιρετικά πιο αργός από τον k -means. Σε κάποια πειράματα παρατηρείται ότι απαιτεί 20 φορές περισσότερες επαναλήψεις, καθώς απαιτείται χρόνος για να περάσει ο αλγόριθμος από το γενικό στάδιο αναζήτησης στο τοπικό.

Παρόλο που ο PSO μπορεί να εκτελεστεί από συστοιχία υπολογιστών παράλληλα, και πάλι είναι αρκετά πιο πολύπλοκος και έχει μεγαλύτερο κόστος σε υπολογιστική ισχύ.

Για να εκμεταλλευτούμε την αποτελεσματικότητα του PSO ως προς την εύρεση καλύτερων κέντρων αλλά και την ταχύτητα του k -means, ειδικά σε δεδομένα με πολλές διαστάσεις άρα και μεγάλη πολυπλοκότητα, μπορούμε να εφαρμόσουμε μια υβριδική μέθοδο κατά την οποία εκτελείται αρχικά ο PSO για μερικές μόνο επαναλήψεις ώστε να κάνει μια γενική αναζήτηση και να καταλήξει σε κάποια βέλτιστα κέντρα κλάσεων. Αυτά τα βέλτιστα κέντρα θα τροφοδοτήσουν το δεύτερο στάδιο της τοπικής αναζήτησης που πραγματοποιείται από τον k -means. Έτσι δεν αφήνουμε στην τύχη την επιλογή των αρχικών κέντρων του k -means αλλά τον τροφοδοτούμε με κάποια από τα βέλτιστα.

Η μέθοδος αυτή είναι πιο αργή σε χρόνο εκτέλεσης από τον k -means αλλά σε ικανοποιητικό επίπεδο αν σκεφτούμε ότι μπορεί να είναι τόσο αποτελεσματική όσο και η εφαρμογή του PSO.

Συνοπτικά ο υβριδικός αλγόριθμος ορίζεται ως εξής:

- Εκκίνηση του PSO για έναν αριθμό επαναλήψεων.
- Ορισμός των κέντρων που βρέθηκαν από τον PSO ως αρχικών κέντρων για τον k -means.
- Εκτέλεση του k -means για έναν αριθμό επαναλήψεων ή όταν εκπληρωθεί μια συνθήκη.

5.5 k -means και Combinatorial PSO

Η υβριδική αυτή μορφή υλοποίησης KCPSO χρησιμοποιεί επίσης τις δυνατότητες και των 2 αλγορίθμων όμως επιπλέον δεν απαιτεί από τον χρήστη να ορίσει εξ αρχής έναν επιθυμητό αριθμό κλάσεων. Αντιθέτως αναλαμβάνει ο ίδιος ο αλγόριθμος να βρει τον βέλτιστο αριθμό κλάσεων με τη

χρήση του CPSO που περιγράψαμε παραπάνω. Έπειτα εκτελώντας τον k-means αναζητά τα καλύτερα κέντρα για τις κλάσεις^[7].

Παρακάτω περιγράφουμε την λειτουργία του:

Για την αναπαράσταση μιας πιθανής λύσης το σωματίδιο αποτελείται από ένα σύνολο ομάδων $\{C_1, C_2, \dots, C_{K_i}\}$ όπου K_i είναι ο αριθμός των κλάσεων που έχει το σωματίδιο i και $\min K_i < K_i < \max K_i$. Η ελάχιστη τιμή είναι συνήθως 2.

Αρχικά κάθε σωματίδιο ορίζει δική του τιμή για το K_i και εφαρμόζει τον k-means για να βρεθεί το αρχικό αποτέλεσμα της ομαδοποίησης. Το αρχικό αποτέλεσμα κάθε σωματίδιο αποτιμάται από την συνάρτηση DB index. Έτσι βρίσκουμε την τιμή του προσωπικού βέλτιστου - p_{best} - κάθε σωματιδίου και του ολικού βέλτιστου - g_{best} . Επίσης το K του ολικού βέλτιστου είναι ο καταλληλότερος αριθμός κλάσεων - K_{best} .

Αναλυτικότερα:

Αρχικά κάνουμε το mapping για να δημιουργήσουμε το διάνυσμα Y .

$$Y_i^t = \begin{cases} 1, & \text{if } K_i^t = K_{gbest}^t \\ -1, & \text{if } K_i^t = K_{ipbest}^t \\ 1 \text{ or } -1 \text{ randomly,} & \text{if } K_i^t = K_{gbest}^t = K_{ipbest}^t \\ 0, & \text{otherwise} \end{cases}$$

Μετά υπολογίζουμε την ταχύτητα και το λ με τους γνωστούς τύπους:

$$V_i^{t+1} = W^{t+1} \times V_i^t + r_1 \times c_1 \times (-1 - Y_i^t) + r_2 \times c_2 \times (1 - Y_i^t)$$

$$\lambda_i^{t+1} = Y_i^t + V_i^{t+1}$$

$$Y_i^{t+1} = \begin{cases} 1, & \text{if } \lambda_i^{t+1} > a^{t+1} \\ -1, & \text{if } \lambda_i^{t+1} < -a^{t+1} \\ 0, & \text{otherwise} \end{cases}$$

και τέλος υπολογίζουμε τη νέα λύση

$$K_i^{t+1} = \begin{cases} K_{gbest}^t, & \text{if } Y_i^{t+1} = 1 \\ K_{ipbest}^t, & \text{if } Y_i^{t+1} = -1 \\ a \text{ random number,} & \text{otherwise} \end{cases}$$

Στο επόμενο στάδιο ελέγχεται η νέα τιμή του K του σωματιδίου και εάν είναι ίδια με αυτή του g_{best} τότε απευθείας αντιγράφει τα κέντρα του g_{best} σαν νέα κέντρα. Για καλύτερη αναζήτηση ένα ή δύο κέντρα επιλέγονται τυχαία και παίρνουν τυχαία τιμή.

Το ίδιο γίνεται και για τα σωματίδια που παίρνουν τον αριθμό κλάσεων από το προσωπικό τους βέλτιστο. Για την περίπτωση που η τιμή του K πρέπει να αλλάξει τυχαία, το σύνολο των κλάσεων επανεκλέγεται σύμφωνα με την νέα τιμή του K . Έτσι αποφεύγονται παγίδες τοπικών βέλτιστων.

Έπειτα αφού έχουν οριστεί ο αριθμός K αλλά και τα διανύσματα των κέντρων, κάθε σωματίδιο εκτελεί τον k -means και υπολογίζονται τα βέλτιστα με βάση την αποτίμηση από την συνάρτηση προσαρμογής.

5.6 PSO και Tabu Search υβριδικός αλγόριθμος

Ο αλγόριθμος “Tabu Search PSO” (TCP SO) συνδυάζει τις μεθόδους του Tabu Search και του PSO πάνω σε ένα σύνολο δεδομένων. Ο λόγος της δημιουργίας του TCP SO είναι το πλεονέκτημα του PSO σε μια πιο γενική αναζήτηση σε ένα μεγάλο εύρος του συνόλου δεδομένων καθώς και το πλεονέκτημα του Tabu Search στην τοπική αναζήτηση.^[22]

Η μέθοδος του TCP SO είναι η εξής:

Βήμα 1: Αρχικοποίηση σωματιδίων του PSO με τυχαίο τρόπο

Βήμα 2: Αποτίμηση της συνάρτησης προσαρμογής του κάθε σωματιδίου

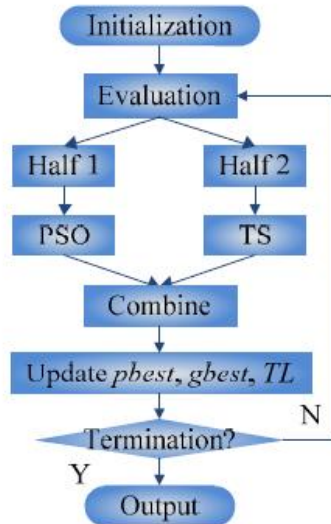
Βήμα 3: Διαχωρισμός των σωματιδίων σε 2 ισοπληθείς ομάδες

Τα σωματίδια της μίας ομάδας θα ανανεώσουν τη θέση τους με τις συναρτήσεις θέσεις και ταχύτητας του PSO ενώ τα σωματίδια της άλλης ομάδας θα ανανεώσουν τη θέση τους με τις συναρτήσεις του Tabu Search αλγορίθμου.

Βήμα 4: Ένωση των 2 ομάδων και καθορισμός του ολικού και του προσωπικού βέλτιστου, καθώς και της λίστας Tabu λύσεων (TL) σύμφωνα με τη μέθοδο του αλγορίθμου Tabu Search, στην οποία αποθηκεύονται υποψήφιες λύσεις που έχουν ήδη επισκεφθεί έως πριν από έναν προκαθορισμένο αριθμό επαναλήψεων.

Βήμα 5: Επανάληψη από το βήμα 2 έως το βήμα 4 και εξαγωγή αποτελέσματος.

Με τη μέθοδο του TCP SO αποφεύγουμε την αργή σύγκλιση του PSO προς βέλτιστη λύση αλλά και το μειονέκτημα της ανεπαρκούς γενικής αναζήτησης του Tabu Search.



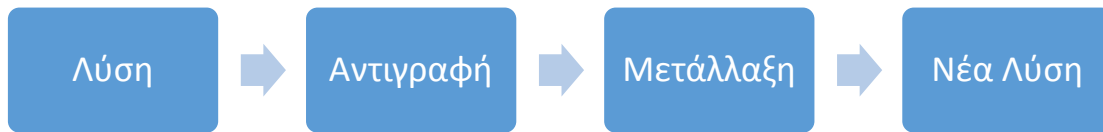
Εικόνα 4 - Διάγραμμα ροής του TCPSO

5.7 Immunity-clonal και PSO με σκοπό την αναγνώριση spam αλληλογραφίας

Προσομοιώνοντας την στρατηγική άμυνας του ανοσοποιητικού συστήματος υλοποιήθηκε ένας νέος αλγόριθμος βασισμένος στον PSO. Όπως τα λεμφοκύτταρα του οργανισμού κλωνοποιούνται και πολλαπλασιάζονται έτσι και στον αλγόριθμο αυτό τα καλύτερα σωματίδια σε διάρκεια κάποιων επαναλήψεων αναπαράγονται με αποτέλεσμα να διευρύνεται ο χώρος αναζήτησης γύρω από μια βέλτιστη υποψήφια λύση και να επιτυγχάνεται η γρηγορότερη σύγκλιση προς τη βέλτιστη λύση. Ο αλγόριθμος αυτός είναι εύκολος στη υλοποίηση, γρηγορότερος και αποτελεσματικότερος στην εύρεση λύσης από τον απλό PSO.^[3]

Με βάση τον απλό PSO τα σωματίδια με τις καλύτερες θέσεις οδηγούν τα υπόλοιπα προς τις βέλτιστες λύσεις. Όμως σε προβλήματα με πολλά τοπικά βέλτιστα υπάρχει ο κίνδυνος παγίδευσης. Σκοπός του clonal PSO είναι η αποφυγή των τοπικών βέλτιστων και η εύρεση της βέλτιστης λύσης με αποτελεσματικότητα.

Η βασική ιδέα του αλγόριθμου είναι να κλωνοποιήσει ένα σωματίδιο σε N , και έπειτα να τα μετασχηματίσει σε N νέες λύσεις.



Εικόνα 5 - Clonal PSO

Περιγραφή του αλγόριθμου

- Κατά την αρχικοποίηση ορίζουμε τις παραμέτρους c_1 , c_2 και α . Επίσης το βάρος να μειώνεται γραμμικά από την τιμή 0.9 σε 0.4.
- Στο δεύτερο βήμα τα σωματίδια ενημερώνουν τη θέση και την ταχύτητά τους σύμφωνα με τις εξισώσεις του βασικού PSO.
- Το βέλτιστο σωματίδιο σύμφωνα με την συνάρτηση προσαρμογής, ορίζεται ως γεννήτορας για την κλωνοποίηση του επόμενου βήματος.
- Έπειτα από M επαναλήψεις, κλωνοποιούνται τα M καλύτερα σωματίδια.
- Στο επόμενο βήμα όλοι οι κλώνοι μεταλλάσσονται ώστε να διαφοροποιηθούν από τους γεννήτορές τους χρησιμοποιώντας για παράδειγμα την μέθοδο Gaussian Noise με την συνάρτηση:

$$P_{gBk} = P_{gBk} + s \times (1-\mu) \times V_{max}$$

- Στο τελευταίο βήμα πριν τον τερματισμό έχουμε την επιλογή των επόμενων λύσεων με τη βοήθεια της συνάρτησης:

$$D(x_i) = (\sum_{j=1}^{N+M} |f(x_i) - f(x_j)|)^{-1}, \quad i = 1, 2, \dots, N+M$$

Από την οποία μπορούμε να υπολογίσουμε την πιθανότητα επιλογής ενός σωματιδίου x_i :

$$p(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{j=1}^{N+M} \frac{1}{D(x_j)}}$$

Ο αλγόριθμος τερματίζει μόλις ικανοποιηθεί κάποια συνθήκη όπως είναι η πραγματοποίηση ενός μέγιστου αριθμού επαναλήψεων.

Βελτιστοποιήσεις του Clonal-PSO

Ο νέος αυτός αλγόριθμος εκμεταλλεύεται πλήρως τον χώρο γύρω από το ολικό βέλτιστο αφού θεωρείται ότι η τελική λύση βρίσκεται σε αυτήν την περιοχή. Σε περίπτωση που βρεθεί τοπικό μέγιστο ο αλγόριθμος έχει δυνατότητα διαφυγής μέσω του μηχανισμού μετάλλαξης και επιλογής αφού υπάρχει πάντα η πιθανότητα επιλογής για σωματίδια που δεν βρίσκονται κοντά στο ολικό βέλτιστο. Στον αλγόριθμο αυτό βλέπουμε ομοιότητες με τους γενετικούς αλγόριθμους.

Παρόλη την αποτελεσματικότητά του υπάρχουν μειονεκτήματα όπως η πολυπλοκότητά του και οι μεγαλύτερες απαιτήσεις σε μνήμη, έτσι έχουν υλοποιηθεί 2 παραλλαγές για τη βελτιστοποίηση του Clonal-PSO.

Πρόοδος και οπισθοδρόμηση (ΠΟ)

Η τεχνική αυτή επιτρέπει στους κλώνους να προχωρούν την αναζήτηση στις τοπικές κοιλάτες αφού θυμάται την πορεία ενός σωματιδίου. Εάν το σωματίδιο συνεχώς βελτιώνει την τιμή της συνάρτησης προσαρμογής τότε συνεχίζει με την ίδια φορά την αναζήτηση ενώ σε αντίθετη περίπτωση οπισθοδρομεί προς την αντίθετη κατεύθυνση.

Η νέα συνάρτηση της ταχύτητας είναι:

$$V_{id}(t+1) = w \times (-\alpha V_{id}(t)) + C_1 \times r_1 \times (P_{iBd}(t) - X_{id}(t)) + C_2 \times r_2 \times (P_{gBd}(t) - X_{id}(t)),$$

όπου $\alpha < 1$.

Ο αλγόριθμος ΠΟ ορίζεται ως εξής:

1. Αρχικοποίηση $u_0 \in [0, \infty)$, $h_0 > 0$, $\alpha > 1$, υπολογισμός $\varphi(u_k)$, $k=0$
2. Σύγκριση τιμών προσαρμογής:
 $u_{k+1} = u_k + h_k$
 $\varphi_{k+1} = \varphi(u_{k+1})$
If $\varphi_{k+1} < \varphi_k$ then go to step 3 else go to step 4
3. Πρόοδος:
 $h_{k+1} = \alpha h_k$
 $u = u_k$
 $u_k = u_{k+1}$
 $\varphi_k = \varphi_{k+1}$
 $k = k + 1$
go to step 2

4. Οπισθοδρόμηση:
if $k = 0$ then $h_k = -h_k$ (αντιστροφή τη κατεύθυνσης της ταχύτητας του σωματιδίου)
 $u_k = u_{k+1}$
go to step 2
5. Έξοδος:
 $\alpha = \min\{u_k, u_{k+1}\}$
 $\beta = \max\{u_k, u_{k+1}\}$

5.8 PSO με μαύρη τρύπα

Με βάση το σκεπτικό της ανίχνευσης μιας μαύρης τρύπας από τους αστρονόμους, κατά το οποίο μια μαύρη τρύπα έλκει τα πάντα γύρω της με αποτέλεσμα να μπορεί να ανιχνευθεί από αυτή την επιρροή στα γειτονικά σώματα εφόσον δεν είναι ορατή με άλλο τρόπο, δημιουργήθηκε η παρακάτω παραλλαγή του clonal-PSO.

Σκοπός της τεχνικής αυτής είναι να δημιουργήσουμε ένα κλώνο κοντά στο βέλτιστο και να τον μεταλλάξουμε με σκοπό να επηρεάσουμε τα υπόλοιπα σωματίδια. Δεν γνωρίζουμε που είναι η τελική λύση αλλά η περιοχή γύρω από το βέλτιστο έχει τη μεγαλύτερη πιθανότητα να το περιέχει.

Κατά τον αλγόριθμο αυτό για κάθε διάσταση σε κάθε επανάληψη επιλέγουμε και κλωνοποιούμε τυχαία ένα σωματίδιο, σε μια ακτίνα r κοντά στο ολικό βέλτιστο, το οποίο θεωρούμε ως μαύρη τρύπα που επηρεάζει τα γύρω σωματίδια και του δίνουμε ένα κατώφλι p στο διάστημα $[0,1]$. Για κάθε διάσταση των υπόλοιπων σωματιδίων δημιουργούμε μια τιμή l . Εάν το l είναι μικρότερο του p τότε το σωματίδιο επηρεάζεται από την μαύρη τρύπα, που σημαίνει ότι η συγκεκριμένη διάσταση του σωματιδίου παίρνει την τιμή της αντίστοιχης διάστασης της μαύρης τρύπας.

Έστω $x_d(t)$ η τιμή της διάστασης d ενός σωματιδίου όπως αυτή θα ήταν με τη συνάρτηση του απλού PSO και $x'_d(t)$ η τιμή της διάστασης d ενός σωματιδίου όπως αυτή θα γίνει με τη χρήση της συνάρτησης της μαύρης τρύπας εάν δηλαδή $l < p$ για την διάσταση d .

Άρα έχουμε τη συνάρτηση:

$$X(t+1) = \begin{cases} XgB + s, & \text{if } l < p \\ x(t+1), & \text{if } l \geq p \end{cases}$$

Ο αλγόριθμος *clonal-PSO* με μαύρη τρύπα:

1. Αρχικοποίηση $c_1=2$, $c_2=2$. Επίσης το βάρος να μειώνεται γραμμικά από την τιμή 0.9 σε 0.4.
2. Υπολογισμός θέσης και ταχύτητας των σωματιδίων με βάση τις συναρτήσεις του απλού PSO.
3. Δημιουργία ενός σωματιδίου κοντά στο βέλτιστο που είναι μαύρη τρύπα σε ακτίνα r . Υπολογισμός s μέσω κανονικής κατανομής στο διάστημα $[-r,r]$.
4. Υπολογισμός του l για κάθε σωματιδίου και αλλαγή της θέσης του με βάση τη σχέση του l με το p .

Σύγκριση του *clonal-PSO* με τις 2 παραλλαγές του

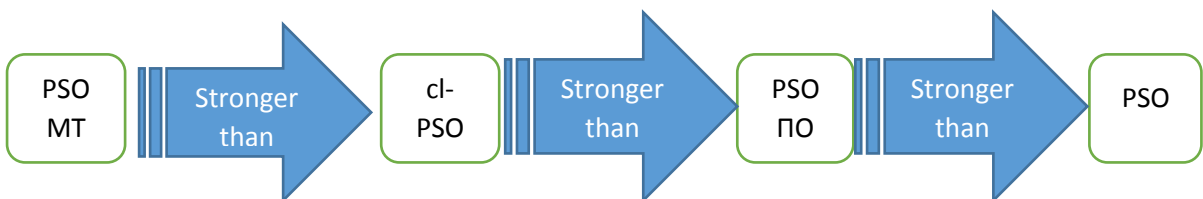
Μέσω της κλωνοποίησης ο χώρος αναζήτησης αυξάνεται σημαντικά με αποτέλεσμα την αύξηση της ταχύτητας σύγκλισης και της αποτελεσματικότητας εύρεσης ολικού βέλτιστου. Συνήθως η παραλλαγή της μαύρης τρύπας προτιμάται ως μια ενδιάμεση λύση που εξισορροπεί την πρόωρη σύγκλιση και την αποτελεσματικότητα στην εύρεση λύσης.

Όσο αφορά την ταχύτητα σύγκλισης η σχέση του βασικού PSO, *clonal-PSO*, PSO με ΠΟ και PSO με μαύρη τρύπα φαίνεται στο παρακάτω γράφημα:



Εικόνα 6 - Σύγκριση ΠΟ/Clonal/MT PSO ως προς την ταχύτητα σύγκλισης

Και όσο αφορά την δυνατότητα εύρεσης ολικού βέλτιστου:



Εικόνα 7 - Σύγκριση ΠΟ/Clonal/MT PSO ως προς την εύρεση ολικού μεγίστου

Εφαρμογή σε αναγνώριση *spam* μηνυμάτων

Για την αναγνώριση *spam* μηνυμάτων έχουν χρησιμοποιηθεί πολλοί αλγόριθμοι και τεχνικές όπως: Naïve Bayes, νευρωνικά δίκτυα και support vector machines (SVM).

Τα SVMs έχουν αποδειχθεί αρκετά αποτελεσματικά στην αναγνώριση προτύπου. Ο αλγόριθμος *clonal-PSO* και οι 2 παραλλαγές του χρησιμοποιούνται στη συγκεκριμένη περίπτωση σε συνδυασμό με τον SVM

κατηγοριοποιητή για την βελτιστοποίηση των παραμέτρων του ώστε να επιτευχθεί και το βέλτιστο αποτέλεσμα. Σαν συνάρτηση προσαρμογής ορίζεται η ακρίβεια κατηγοριοποίησης με χρήση της τεχνικής 10-fold cross validation που έχει ως εξής:

1. Διαχωρισμός δεδομένων πλήθους n σε 10 ομάδες μεγέθους $n/10$.
2. Εκπαίδευση σε 9 ομάδες και τεστ σε 1.
3. Επανάληψη 10 φορές και σαν αποτέλεσμα παίρνουμε τον μέσο όρο των αποτελεσμάτων των επαναλήψεων.

Μετά από μια σειρά πειραμάτων ο clonal-PSO και οι 2 παραλλαγές αποδεικνύονται πολύ αποτελεσματικοί στην αναγνώριση spam μηνυμάτων τόσο στην ακρίβεια της λύσης όσο και στην ταχύτητα εύρεσής της.

5.9 Δυαδικός PSO και αναγνώριση spam μηνυμάτων

Ο αλγόριθμος PSO μπορεί να χρησιμοποιηθεί για να αναπαραστήσει δυαδικές λύσεις πέρα από συνεχείς.

Στον δυαδικό PSO η συνάρτηση ταχύτητας παραμένει ίδια αλλά η συνάρτηση θέσης αλλάζει και ορίζεται ως εξής.^{[21] [20]}

$$X_i^{t+1} = \begin{cases} 1, & \text{if } rand_i^{t+1} < sig(V_i^{t+1}) \\ 0, & \text{otherwise} \end{cases}$$

Η παράμετρος $rand$ είναι ένα διάνυσμα τυχαίων αριθμών από την κανονική κατανομή στο διάστημα $[0,1]$ και η συνάρτηση $sig()$ είναι η σιγμοειδής συνάρτηση.

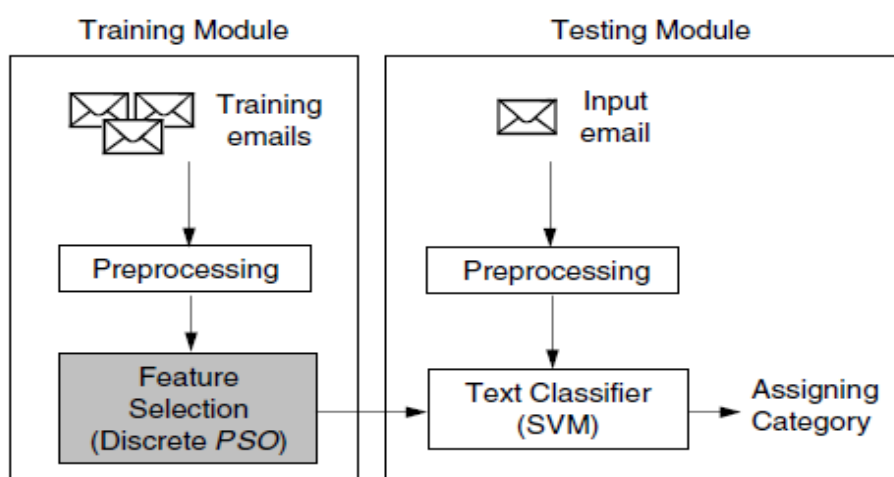
Κατά την αναγνώριση spam μηνυμάτων σκοπός μας είναι η κατηγοριοποίηση μηνυμάτων σε 2 κατηγορίες, spam και not-spam οι οποίες μπορούν να χαρακτηριστούν από τις τιμές 1 και 0 γι αυτό και είναι κατάλληλη η χρήση του δυαδικού PSO. Τα μηνύματα συνήθως αναπαρίστανται σαν διανύσματα όρων και έπειτα από μια διαδικασία προεπεξεργασίας όπως η αφαίρεση άρθρων και συνδέσμων, η απαλοιφή μη χρήσιμων όρων ή η αναγωγή των λέξεων στις ριζικές λέξεις τους (απαλοιφή καταλήξεων πχ ανεβ-αίνω, ανεβ-αίνεις, ανεβ-ασμα) παρέχονται σε έναν κατηγοριοποιητή, συνήθως SVM, για να αναγνωριστούν.

Έχουμε αποτελεσματικότερη κατηγοριοποίηση όταν ο αριθμός αυτών των όρων είναι ο βέλτιστος γιατί μειώνουμε την πολυπλοκότητα και τον χρόνο.

Η δουλειά της εύρεσης των βέλτιστων όρων πραγματοποιείται από τον αλγόριθμο PSO ο οποίος μπορεί να αναπαραστήσει πολλές λύσεις ταυτόχρονα μέσω των σωματιδίων και να τις αξιολογήσει βρίσκοντας την βέλτιστη.

Η συγκεκριμένη τεχνική χρησιμοποιεί ένα σύνολο μηνυμάτων για εκπαίδευση μέσω του PSO ώστε να βρεθεί το βέλτιστο πλήθος όρων που θα περιέχει ένα διάνυσμα αναπαράστασης μηνύματος.

Στο παρακάτω σχεδιάγραμμα βλέπουμε την ροή της τεχνικής που περιγράψαμε:



Εικόνα 8 - Διαδικασία αναγνώρισης spam μηνυμάτων από τον δυαδικό PSO

5.10 PSO με χρήση ρόλων στα σωματίδια για ομαδοποίηση εγγράφων Ιστού

Ο αλγόριθμος RolePSO δημιουργήθηκε για την βελτίωση του απλού PSO ως προς το θέμα της πρόωρης σύγκλισης σε τοπικά βέλτιστα. Για τον σκοπό αυτό χωρίζει το σμήνος των σωματιδίων σε 3 ομάδες, οι οποίες λειτουργούν με διαφορετικό στόχο και ονομάζονται ρόλοι.^[4]

Στον απλό PSO όλα τα σωματίδια κινούνται προς το τρέχον βέλτιστο με την ταχύτητα που ορίζει η, κοινή για όλα, συνάρτηση του PSO. Αυτό ακριβώς μπορεί να οδηγήσει σε πρόωρη σύγκλιση προς ένα τοπικό βέλτιστο. Στον RolePSO όμως, κάθε μία από τις 3 ομάδες/ρόλους έχει μια διαφορετική εξίσωση η οποία είναι κοινή μόνο για τα σωματίδια του ίδιου ρόλου. Αυτό που διαφοροποιείται από ρόλο σε ρόλο είναι η τιμή της αδράνειας w.

Ο κανόνας για την τιμή της αδράνειας των 3 διαφορετικών ρόλων έχει ως εξής:

- **Ρόλος 1:** Η τιμή της αδράνειας είναι μεγάλη, με αποτέλεσμα τα σωματίδια του ρόλου αυτού να πραγματοποιούν μια ευρύτερη αναζήτηση του χώρου ώστε να βρεθεί το ολικό βέλτιστο.
- **Ρόλος 2:** Η τιμή της αδράνειας μειώνεται γραμμικά όπως στον απλό PSO.
- **Ρόλος 3:** Η τιμή της αδράνειας παίρνει μικρή τιμή με σκοπό να είναι πιο αποτελεσματική η τοπική αναζήτηση.

Με αυτή τη νέα τροποποίηση η συνάρτηση της ταχύτητας ορίζεται ως εξής:

$$V_i(t+1) = W'_{role} \times V_i(t) + C_1 \times U(0,1) \otimes (B_i(t) - X_i(t)) + C_2 \times U(0,1) \otimes (B_g(t) - X_i(t))$$

όπου W'_{role} είναι η αδράνεια του κάθε ρόλου και $role = 1,2,3$.

Ενδεικτικά για τον πρώτο ρόλο η τιμή καθορίζεται συνήθως στο 0.72, για τον δεύτερο ρόλο η τιμή μειώνεται με την συνάρτηση $w - (iterationNo - 1) \times 0.01 \times w$ και για τον τρίτο ρόλο η τιμή μειώνεται εκθετικά με τη συνάρτηση $0.72 / \exp((iterationNo - 1) / maxIterationNo)$.

Ο αλγόριθμος rolePSO

- Για κάθε σωματίδιο p_i
 - Υπολογισμός συνάρτησης προσαρμογής $f(p_i)$
 - If $f(p_i) < pBestValue_i$ then $pBestValue_i = f(p_i)$ και $pBestParticle_i = p_i$
Else χρήση κατάλληλης συνάρτησης για την ανανέωση πρώτα της αδράνειας $wgrou_p_i$ και έπειτα της ταχύτητας και της θέσης του p_i .
- Αναζήτηση του $globalBestValue$ και ανάθεση του σωματιδίου ως $globalBestParticle$
- Τερματισμός μετά από 30 επαναλήψεις

Η εφαρμογή στα έγγραφα Ιστού

Για τον έλεγχο της απόδοσης του αλγόριθμου έχει πραγματοποιηθεί ένα πείραμα επιβλεπόμενης μάθησης. Χρησιμοποιήθηκαν 15 σωματίδια για 30 επαναλήψεις.

Ο rolePSO εφαρμόστηκε στα εξής datasets:

- Bekkerman name dataset (αποτελέσματα 100 σελίδων του google)
- Wikipedia (αποτελέσματα 100 σελίδων του yahoo)
- ECDL person name (αποτελέσματα 100 σελίδων του yahoo)

Αρχικά τα έγγραφα επεξεργάστηκαν ώστε να αφαιρεθούν οι λέξεις χωρίς σημασία όπως τα άρθρα κλπ και δημιουργήθηκε ο πίνακας των εγγράφων A .

Κάθε στοιχείο A_{ij} υπολογίζεται ως $A_{ij} = TF_{ij} \times IDF_{ij}$

Μετά δημιουργούμε τον γράφο εγγράφων G με τη συνάρτηση

$$G = A^T \times A(i,j) = \sum_{k \in \{TermSet\}} d_{ik} \times d_{jk}$$

Το στοιχείο $G(i,j)$ συμβολίζει την ομοιότητα 2 εγγράφων d_i και d_j . Μια μεγάλη τιμή του G δείχνει ότι τα 2 έγγραφα έχουν πολλούς κοινούς όρους.

Η συνάρτηση προσαρμογής που χρησιμοποιήθηκε είναι η αναλογία ομοιότητας στις ίδιες κλάσεις ως προς την διαφορετικότητα ανάμεσα στις κλάσεις:

$$f = \frac{\sum_{k=1}^N \sum_{x_i \in C_k} (x_i - m_k)^2 / C_k}{\sum_{i,j \in C} (m_i - m_j)^2}$$

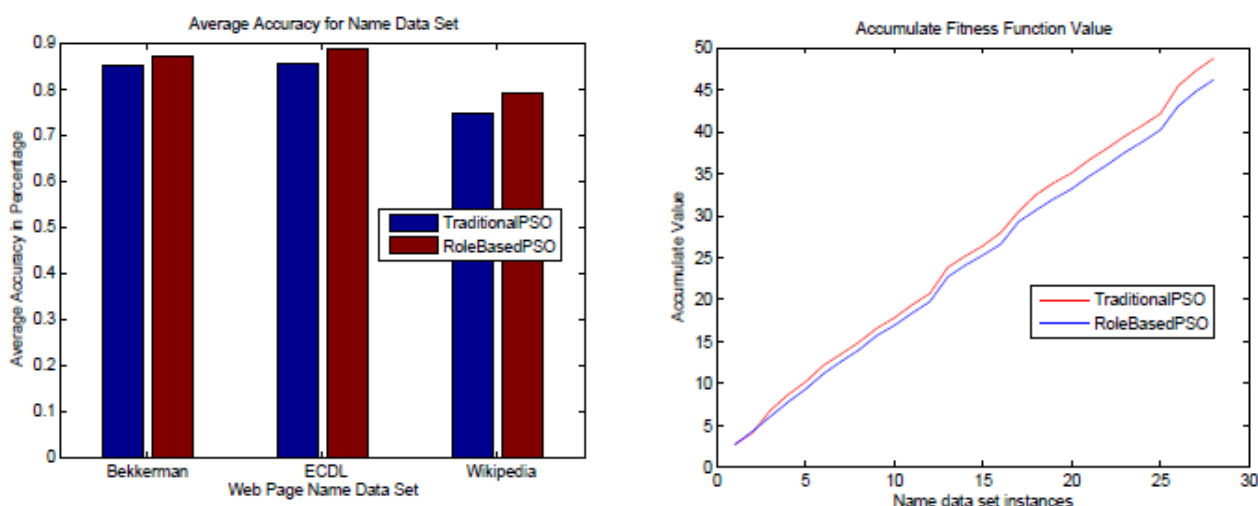
Τέλος για τη μέτρηση της απόδοσης της διαδικασίας χρησιμοποιήθηκε η συνάρτηση ακρίβειας:

$$AC_i = \frac{CorrectlyPredict_i}{Predict_i}$$

σε συνδυασμό πάλι με την συνάρτηση προσαρμογής για τη μέτρηση της ομοιότητας και ανομοιότητας.

Στα αποτελέσματα παρατηρήθηκε βελτιστοποίηση στην ακρίβεια και στην ταχύτητα σχετικά με τον απλό PSO. Συγκεκριμένα για την ακρίβεια ο rolePSO ήταν πιο ακριβής κατά 2.5% στο Bekkerman dataset, 3% στο ECDL και 5% στη Wikipedia.

Σχετικά με την ταχύτητα ο rolePSO βρίσκει το βέλτιστο κατά 5% γρηγορότερα από τον απλό PSO. Τα παραπάνω αποτελέσματα φαίνονται αντίστοιχα στα 2 παρακάτω γραφήματα:



Εικόνα 9 - Συγκριση βασικού PSO και PSO με ρόλους

Σαν γενική παρατήρηση ο rolePSO αποδίδει καλύτερα όταν τα δεδομένα των εγγράφων είναι δυσκολότερο να ομαδοποιηθούν.

5.11 Εκπαίδευση πιθανοτικού νευρωνικού δικτύου με τη χρήση PSO

Παρακάτω περιγράφεται μια εφαρμογή του PSO στη δημόσια διοίκηση σχετικά με τις οικονομικές δραστηριότητες των επιχειρήσεων που είναι δηλωμένες σε αυτήν.^[11]

Κάθε εταιρεία κατά την ίδρυσή της συντάσσει ένα καταστατικό το οποίο περιγράφει μεταξύ άλλων και τις δραστηριότητες της. Ενώ οι οικονομικοί φορείς ενός κράτους κατατάσσουν κάθε εταιρεία σε κατηγορίες δραστηριοτήτων.

Η εργασία αυτή γίνεται από υπαλλήλους που ξοδεύουν πολύ χρόνο για την κατηγοριοποίηση και τον έλεγχο σφαλμάτων κατά τη διαδικασία.

Σκοπός της εφαρμογής είναι η ανάθεση εταιρειών σε κατηγορίες δραστηριοτήτων βάσει του καταστατικού τους. Για την εργασία αυτή χρησιμοποιείται ως κατηγοριοποιητής ένα πιθανοτικό νευρωνικό δίκτυο (probabilistic neural network - PNN).

Η μοντελοποίηση του προβλήματος έχει ως εξής:

$C = \{c_1, c_2, \dots, c_{|C|}\}$ είναι οι προκαθορισμένες κατηγορίες δραστηριοτήτων

$W = \{d_1, d_2, \dots, d_{|W|}\}$ είναι το σύνολο των εγγράφων που κατηγοριοποιήθηκαν από ανθρώπους

$TV = \{d_1, d_2, \dots, d_{|TV|}\}$ είναι το σύνολο των εγγράφων που έχουν κατηγοριοποιηθεί

$Te = \{d_{|TV|+1}, d_{|TV|+2}, \dots, d_{|W|}\}$ είναι το σύνολο των εγγράφων που οι κατηγορίες τους είναι άγνωστες.

Το σύνολο TV χρησιμεύει για εκπαίδευση του συστήματος ενώ το Te για την πρόβλεψη από το PNN.

$f(d_j, c_i)$ είναι η αποτίμηση της λύσης όταν το έγγραφο d_j κατατάσσεται στην κατηγορία c_i .

$r(d_j, c_i)$ είναι ο βαθμός καταλληλότητας μιας κατηγορίας για κάποιο έγγραφο. Μεγαλύτερο f σημαίνει μικρότερο r . Πολλές φορές μόνο οι κατηγορίες με μικρότερο r (πιο κατάλληλες) ανατίθενται σε ένα έγγραφο, όταν $f(d_j, c_i) > \text{threshold } \tau$.

Κάθε έγγραφο που κατηγοριοποιείται αποτιμάται από τις συναρτήσεις $hloss$, $one-error$, $ranking-loss$, $coverage$ και $average\ precision$. Η γενική αποτίμηση γίνεται με τον υπολογισμό του μέσου όρου κάθε συνάρτησης, για παράδειγμα

$$hloss = \frac{1}{p} \times \sum_{j=1}^p hloss_j$$

Το πιθανοτικό νευρωνικό δίκτυο

Ένα ΠΝΔ υπολογίζει την έξοδο μιας εισόδου x με την συνάρτηση:

$$F_{k,i}(x) = \frac{1}{2\pi\sigma^2} \times \exp\left(-\frac{x^t w_{ki} - 1}{\sigma^2}\right)$$

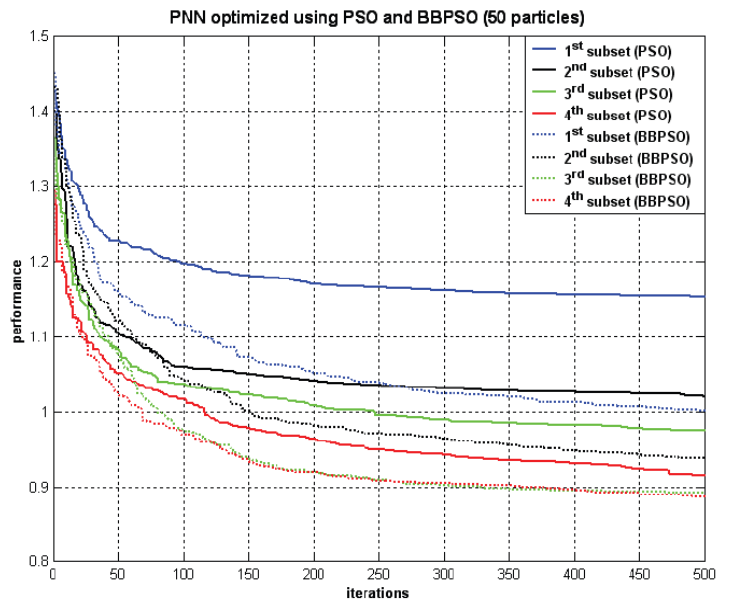
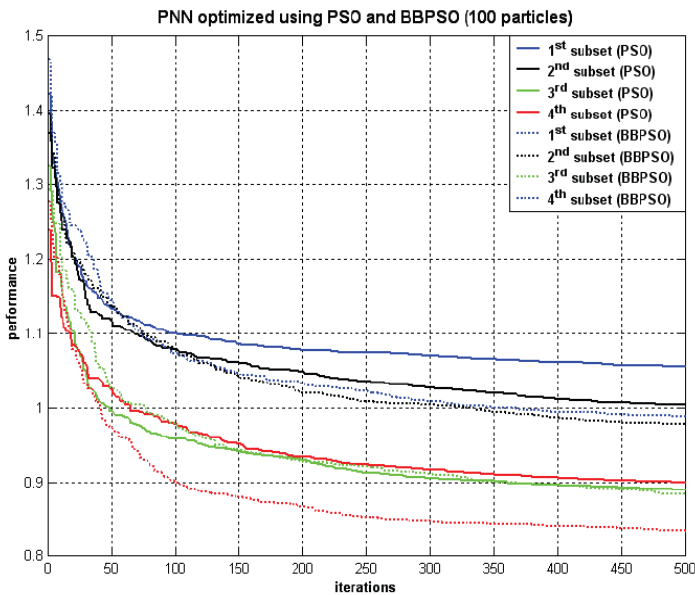
όπου x το διάνυσμα εισόδου, w_{ki} είναι το k -οστό δείγμα ενός νευρώνα κλάσης C_i και σ είναι η Gaussian σταθερά απόκλισης της Gaussian καμπύλης.

Η παράμετρος σ είναι αυτή που πρέπει να βελτιστοποιηθεί από τον αλγόριθμο PSO.

Αυτό που αλλάζει στον νέο αλγόριθμο είναι ότι καταργείται η συνάρτηση της ταχύτητας και η συνάρτηση θέσης διαμορφώνεται ως εξής:

Για κάθε σωματίδιο i , $N = (\mu_i, \sigma_i^2)$

όπου $\mu_i = \frac{g_{best} + p_{best}}{2}$ και $\sigma_i = |g_{best} - p_{best}|$ και N είναι η Gaussian κατανομή.



Εικόνα 10 - Σύγκριση απόδοσης PNN που εκπαιδεύτηκε με PSO και BBPSO των 50 και 100 σωματιδίων

Βλέποντας την παραπάνω περίπτωση καταλαβαίνουμε ότι ο αλγόριθμος PSO μπορεί να χρησιμοποιηθεί εμμέσως σε προβλήματα ομαδοποίησης, δηλαδή ως μέσο αποτελεσματικότερης εκπαίδευσης άλλων αλγορίθμων ή την επιλογή των καταλληλότερων χαρακτηριστικών από ένα σύνολο δεδομένων.

5.12 Ομαδοποίηση εικόνων με τη χρήση PSO

Το πρόβλημα της ομαδοποίησης των pixels των εικόνων μοντελοποιείται ως εξής^{[8][9][10]}:

- N_c ο αριθμός των κλάσεων
- $x_i = (m_{i1}, \dots, m_{ij}, \dots, m_{iN_c})$ είναι το διάνυσμα ενός σωματιδίου με m_{ij} να δηλώνει το διάνυσμα του κέντρου της j κλάσης αναφορικά με το i σωματίδιο. Δηλαδή το κάθε σωματίδιο αναπαριστά μια πιθανή ομαδοποίηση.
- Συνάρτηση προσαρμογής $f(x_i, Z) = w_1 d_{\max}(Z, x_i) + w_2 (Z_{\max} - d_{\min}(x_i))$ όπου $Z_{\max} = 2^s - 1$ η μέγιστη τιμή ενός pixel για μια εικόνα με s bits.
- Πίνακας Z του οποίου κάθε στοιχείο Z_{ijp} δηλώνει εάν το pixel Z_p ανήκει στην κλάση C_{ij} του σωματιδίου i .
- w_1, w_2 σταθερές ορισμένες από τον χρήστη.
- $d_{\max}(Z, x_i) = \max_{j=1, \dots, N_c} \{ \sum_{\forall Z_p \in C_{ij}} d(Z_p - m_{ij}) / |C_{ij}| \}$

η μέγιστη ευκλείδεια απόσταση ανάμεσα στα κέντρα των κλάσεων και των αντίστοιχών τους pixels, ενός σωματιδίου i .

- $d_{\min}(x_i) = \min_{j_1 \neq j_2} \{ m_{ij_1} - m_{ij_2} \}$

η ελάχιστη ευκλείδεια απόσταση κάθε ζεύγους κέντρων του σωματιδίου i .

Η εφαρμογή του αλγόριθμου

1. Αρχικοποίηση των σωματιδίων ώστε να περιέχουν N_c τυχαία κέντρα.
2. Για κάθε σωματίδιο i :
 - a. Για κάθε pixel z_p :
 - i. Υπολογισμός $d(Z_p - m_{ij})$ για κάθε κλάση C_{ij}
 - ii. Ανάθεσε το z_p στην κλάση C_{ij} όπου
$$d(Z_p - m_{ij}) = \min_{c=1, \dots, N_c} \{d(z_p - m_{ic})\}$$
 - b. Υπολογισμός προσαρμογής $f(x_i(t), Z)$
3. Εύρεση ολικού βέλτιστου
4. Αλλαγή των κέντρων των σωματιδίων με τη χρήση των συναρτήσεων του PSO.
5. Τερματισμός όταν ικανοποιηθεί η συνθήκη τερματισμού.



Εικόνα 11 - Αναπαράσταση των ομαδοποιημένων pixels σε μια εικόνα μαγνητικής τομογραφίας εγκεφάλου, με χρήση του PSO.

5.13 Μη Επιβλεπόμενη ομαδοποίηση εικόνων με δυναμική ομαδοποίηση PSO (DCPSO)

Η τεχνική της δυναμικής ομαδοποίησης PSO (dynamic clustering PSO) μπορεί να χρησιμοποιηθεί για την εύρεση προτύπων σε εικόνες.

Η τεχνική αυτή ακολουθεί την ομαδοποίηση διαμερισμού (partitional clustering), όμως απαιτεί ελάχιστη είσοδο από τον χρήστη καθώς μπορεί να καθορίσει αυτόματα τον βέλτιστο αριθμό κλάσεων για ένα δεδομένο σύνολο δεδομένων.

Ο αλγόριθμος αρχικά ξεκινάει διαχωρίζοντας τα δεδομένα σε έναν μεγάλο αριθμό κλάσεων και σταδιακά, με τη χρήση του δυαδικού PSO που εξηγήσαμε παραπάνω, επαναπροσδιορίζει τον αριθμό των κλάσεων μέχρι να καταλήξει στον βέλτιστο. Στη συνέχεια τα κέντρα των κλάσεων ρυθμίζονται από τον αλγόριθμο k-means και στις επόμενες επαναλήψεις εφαρμόζεται ξανά ο δυαδικός PSO. [10][9][8]

Ο αλγόριθμος DCPSO

Οι βασικοί όροι του αλγόριθμου είναι:

- N_c ο μέγιστος αριθμός κλάσεων
- N_d οι διαστάσεις των δεδομένων
- N_p ο αριθμός των διανυσμάτων που πρέπει να ομαδοποιηθούν
- $Z = \{z_{j,p} \in R \mid j = 1, \dots, N_d \mid \text{και } p = 1, \dots, N_p\}$ είναι το σύνολο των δεδομένων.
- $M = \{m_{j,k} \in R \mid j = 1, \dots, N_d \mid \text{και } k = 1, \dots, N_c\}$ είναι το σύνολο των κέντρων των N_c κλάσεων.
- $S = \{x_1, \dots, x_i, \dots, x_s\}$ είναι το σμήνος των s σωματιδίων όπου x_i δηλώνει το σωματίδιο i , με $x_{i,k} \in \{0,1\}$ για $k = 1, \dots, N_c$ έτσι ώστε εάν $x_{i,k} = 1$ τότε το αντίστοιχο m_k έχει επιλεγθεί στην προτεινόμενη από το x_i λύση, αντιθέτως εάν $x_{i,k} = 0$ τότε το m_k δεν έχει επιλεγθεί.
- n_i είναι ο αριθμός των κλάσεων που έχει επιλεγθεί από τη λύση του σωματιδίου x_i .

$$n_i = \sum_{k=1}^{N_c} x_{i,k}, n_i \leq N_c$$

δηλαδή είναι ίσο με τα κέντρα που έχουν επιλεγθεί ως λύσεις από το σωματίδιο x_i .

- M_i είναι η λύση που έχει επιλεγθεί από το σωματίδιο x_i , ώστε $M_i = (m_k) \forall k: x_{i,k} = 1 \text{ με } M_i \subseteq M$.
- n_T είναι ο αριθμός των κλάσεων που ορίζονται από την λύση που προτείνει το ολικό βέλτιστο σωματίδιο y' .

$$n_{\tau} = \sum_{k=1}^{N_c} y'_{k} , n_{\tau} \leq N_c$$

- M_{τ} είναι η λύση που έχει επιλεγθεί από το ολικό βέλτιστο σωματίδιο y' , ώστε $M_{\tau} = (m_k) \forall k: y'_{k} = 1 \text{ με } M_{\tau} \subseteq M$.
- M_r είναι το σύνολο των κέντρων των κλάσεων που δεν έχουν επιλεγθεί από τη λύση που αναπαριστά το ολικό βέλτιστο σωματίδιο y' , ώστε $M_r = (m_k) \forall k: y'_{k} = 0 \text{ με } M_r \subseteq M$.
- p_{ini} είναι μια τιμή πιθανότητας που χρησιμοποιείται στην αρχικοποίηση της θέσης των σωματιδίων, η οποία ορίζεται από τον χρήστη. Η αρχικοποίηση δίνεται από τη συνάρτηση:

$$x_{i,k} = \begin{cases} 0, & \text{αν } r_k(t) \geq p_{ini} \\ 1 & \text{αν } r_k(t) < p_{ini} \end{cases}$$

όπου r_k τυχαία τιμή από την κανονική κατανομή στο διάστημα (0,1). Μια μεγαλύτερη τιμή στο p_{ini} δίνει περισσότερες πιθανότητες στα κέντρα του M να επιλεγθούν.

Το κάθε σωματίδιο αναπαριστά με 0 ή 1 εάν θα χρησιμοποιήσει κάθε μία από τις N_c κλάσεις στην λύση που αναπαριστά. Τα pixels ανατίθενται στην κοντινότερη ομάδα τους και αποτιμάται η λύση. Έπειτα το particle με τη χρήση των δυαδικών συναρτήσεων ταχύτητας και θέσης αλλάζει τις επιλεγόμενες ομάδες και η εκτέλεση του PSO επαναλαμβάνεται. Τέλος ο k-means προσαρμόζει τη βέλτιστη λύση και όλη η διαδικασία επαναλαμβάνεται.

Περιγραφή και ανάλυση του αλγόριθμου

1. Επιλογή τυχαίων m_k κέντρων για να σχηματιστεί το σύνολο M από τα σημεία του Z .
2. Αρχικοποίηση του σμήνους S με βάση τη συνάρτηση που περιγράψαμε παραπάνω.
3. Αρχικοποίηση της ταχύτητας v_i των σωματιδίων με τυχαίες τιμές κάθε διάστασης στο διάστημα [-5,5].
4. Για κάθε σωματίδιο x_i :
 - a. Διαχωρισμός του Z σύμφωνα με τα κέντρα του M_i με ανάθεση κάθε z_p στην κοντινότερη κλάση του M_i .
 - b. Υπολογισμός της συνάρτησης προσαρμογής $f(x_i)$.
5. Εφαρμογή εξισώσεων του δυαδικού PSO για την αλλαγή της θέσης και της ταχύτητας των σωματιδίων.
6. Επανάληψη των βημάτων 4 και 5.
7. Επαναπροσδιορισμός του M_{τ} με την εφαρμογή του k-means.
8. Τυχαία επιλογή του M_r από το Z .
9. Όρισε $M = M_{\tau} \cup M_r$
10. Επανάληψη των βημάτων 2 έως 9.

Ο αλγόριθμος DCPSO επιλέγει αρχικά έναν τυχαίο αριθμό κέντρων από το Z και σχηματίζει το M και το σμήνος S αρχικοποιείται. Μετά εφαρμόζεται ο δυαδικός PSO για να βρεθεί το βέλτιστο σύνολο M_T κέντρων και εφαρμόζεται ο k-means για επαναπροσδιορισμό τους. Στη συνέχεια λαμβάνεται ως νέο M η ένωση του M_T με το M_r μαζί με κάποια ακόμα τυχαία επιλεγμένα σημεία του Z για διαφοροποίηση. Στην επόμενη επανάληψη ο αλγόριθμος εκτελείται και πάλι με βάση το νέο M, μέχρι να ικανοποιηθούν ορισμένα κριτήρια.

Συνάρτηση Προσαρμογής

Σαν συνάρτηση προσαρμογής χρησιμοποιείται η:

$$F = (c \times N(2,1) + 1) \times \frac{inter}{intra}$$

Όπου N είναι η Gaussian κατανομή με μέσο = 2 και απόκλιση = 1, $inter = \frac{1}{N_p} \sum_{k=1}^K \sum_{u \in C_k} \|u - m_k\|^2$, είναι η απόσταση κάθε σημείου κάθε κλάσης από το κέντρο της αντίστοιχής του κλάσης, $inter = \min\{\|m_k - m_{kk}\|^2\}$ είναι η ελάχιστη απόσταση των κέντρων μεταξύ τους και c είναι παράμετρος που ορίζεται από τον χρήστη.

Ο αλγόριθμος DCPSO έχει χρησιμοποιηθεί δίνοντας πολύ καλά αποτελέσματα, στην ομαδοποίηση pixels εικόνων από μαγνητικές τομογραφίες και δορυφόρους.

6. ΥΛΟΠΟΙΗΣΗ ΜΕ ΒΑΣΗ ΤΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Στην εργασία αυτή υλοποιήσαμε τον αλγόριθμο PSO με βάση αυτά που μελετήσαμε στην βιβλιογραφία έτσι ώστε να πραγματοποιήσουμε και τα πειράματα που παρουσιάζουμε στο επόμενο κεφάλαιο.

Για την υλοποίηση του αλγόριθμου βασιστήκαμε στη βιβλιοθήκη JSWARM-PSO(<http://jswarm-pso.sourceforge.net>), την έκδοση δηλαδή του αλγόριθμου σε γλώσσα προγραμματισμού JAVA.

Η παραπάνω βιβλιοθήκη είναι η υλοποίηση του αλγορίθμου μόνο για προβλήματα βελτιστοποίησης όπως για παράδειγμα την εύρεση λύσης για την ελαχιστοποίηση ή μεγιστοποίηση της τιμής μιας συνάρτησης. Ο αλγόριθμος θεωρούσε κάθε διάσταση ενός σωματιδίου ως μια μονοδιάστατη λύση (π.χ κάθε διάσταση ήταν και μία μεταβλητή της συνάρτησης).

Όπως είναι προφανές η υλοποίηση έπρεπε να επανεξεταστεί και να επεκταθεί για να προσαρμοστεί σε προβλήματα ομαδοποίησης (επιβλεπόμενης και μη επιβλεπόμενης). Επίσης με την προσαρμογή αυτή έπρεπε να υλοποιηθεί από την αρχή ένα μεγάλο κομμάτι της εφαρμογής, λειτουργίες δηλαδή σχετικές με την ομαδοποίηση δεδομένων, τις δυσκολίες με την ανομοιότητα δεδομένων, κάποιες χρήσιμες παραλλαγές του PSO κ.α

Κατά την υλοποίηση του PSO δημιουργήθηκαν οι εξής επεκτάσεις και διορθώσεις της εφαρμογής (για την αποφυγή παρερμηνείας του όρου «κλάση» ενός σωματιδίου με τον όρο «κλάση» της JAVA, θα αναφέρουμε τις κλάσεις της Java με τον όρο «αρχείο»):

6.1 Υλοποίηση μηχανισμού για φόρτωση των δεδομένων από αρχεία ARFF.

Όπως παρουσιάζουμε στο κεφάλαιο 8 χρησιμοποιήσαμε αρχεία ARFF από το WEKA project σαν σύνολα δεδομένων για την υλοποίησή μας. Το WEKA project παρέχει βιβλιοθήκες JAVA με διάφορες συναρτήσεις που μπορεί κανείς να χρησιμοποιήσει σε εφαρμογές.

Στο αρχείο **Dataset.java**, υλοποιήσαμε την μέθοδο **buildDataset()** η οποία φορτώνει τα δεδομένα από αρχεία ARFF με τη χρήση της βιβλιοθήκης του WEKA καθώς επίσης προσδιορίζει και τον δείκτη της κλάσης των δεδομένων. Τέλος η ίδια συνάρτηση προσδιορίζει τον αριθμό των διαστάσεων των δεδομένων και τον αριθμό των κλάσεων.

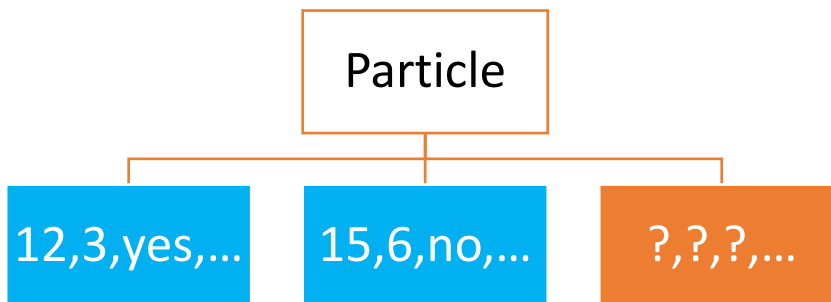
6.2 Επανασχεδιασμός της κλάσης των σωματιδίων ώστε κάθε θέση του να είναι μια πολυδιάστατη υποψήφια λύση.

Η βιβλιοθήκη JSWARM-PSO πάνω στην οποία βασιστήκαμε βασιζόταν στο ότι κάθε υποψήφια λύση είναι μονοδιάστατη. Αυτή ήταν μια βασική αλλαγή που έπρεπε να κάνουμε. Έτσι υλοποιήσαμε την αναπαράσταση των κλάσεων (υποψήφιων λύσεων) μέσα σε στα σωματίδια ως πολυδιάστατα αντικείμενα.

Αυτή η αλλαγή δεν βρίσκεται μέσα σε κάποια συγκεκριμένη κλάση ή συνάρτηση αλλά εφαρμόστηκε σε όλα τα σημεία που έπρεπε να αλλάξουν σύμφωνα με τα νέα δεδομένα. Για παράδειγμα στο αρχείο **ClassificationFitness.java** και στο **ClusteringFitness.java** η συνάρτηση προσαρμογής για επιβλεπόμενη και μη επιβλεπόμενη ομαδοποίηση αντίστοιχα, υλοποιήθηκε από την αρχή ώστε να υποστηρίζει τα πολυδιάστατα δεδομένα και να γίνονται οι συγκρίσεις της θέσης των δεδομένων με τις αντίστοιχες κλάσεις. Ακόμα στο **TestPhase.java** πραγματοποιείται η ομαδοποίηση του PSO μετά τη φάση της εκπαίδευσης. Εκεί γίνεται η αποτίμηση της διαφοράς των δεδομένων με κάθε κλάση ώστε να προσδιοριστεί η κλάση με τη μικρότερη διαφορά και να γίνει η ανάθεση σε αυτήν.

6.3 Υλοποίηση δυναμικού καθορισμού των διαστάσεων και του αριθμού των κλάσεων ενός σωματιδίου (αναλόγως με τις διαστάσεις των δεδομένων)

Ο υλοποίηση του JSWARM-PSO, εφόσον τα δεδομένα ήταν μονοδιάστατα, δημιουργούσε στατικά τις διαστάσεις των σωματιδίων. Καθώς η υλοποίησή μας υπολογίζει δυναμικά τον αριθμό των ομάδων αλλά και τις διαστάσεις κάθε ομάδας, έτσι στο αρχείο **ClassificationParticle.java** (αντίστοιχα **ClusteringParticle.java**) μπορούμε να δούμε ότι ένα σωματίδιο δημιουργείται υπολογίζοντας τις διαστάσεις τη στιγμή της δημιουργίας, δυναμικά, χωρίς να χρειάζεται προκαθορισμός από τον χρήστη.



Εικόνα 12 Δυναμική ανάθεση διαστάσεων και ομάδων

6.4 Υλοποίηση βοηθητικών μεθόδων για εύρεση μέγιστης και ελάχιστης τιμής κάθε διάστασης στα δεδομένα.

Στο αρχείο **Utils.java** έχουμε δύο συναρτήσεις:

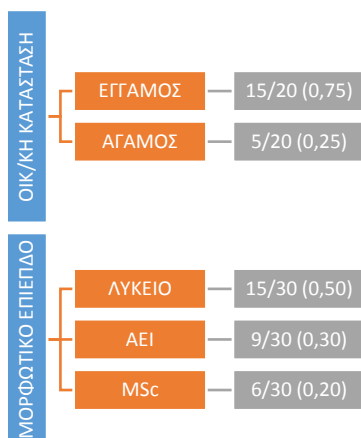
1. minValues()
2. maxValues()

Οι 2 πρώτες δημιουργούν ένα πίνακα στη μνήμη με την μέγιστη τιμή και έναν με την ελάχιστη τιμή, αντίστοιχα, που μπορεί να πάρει κάθε διάσταση των δεδομένων. Για τα μη αριθμητικά δεδομένα η μέγιστη τιμή είναι πάντα 1 και η ελάχιστη 0, σύμφωνα με την υλοποίηση της αντιστοίχισής τους με αριθμητικές τιμές.

6.5 Υλοποίηση διερμηνείας των μη αριθμητικών τιμών σε τιμές πραγματικών αριθμών.

Επιλέξαμε την επίλυση του προβλήματος των nominal τιμών όπως προτείνεται από τους Νουαουρία-Βουκαδουμ^[25] Δηλαδή κάθε τέτοια τιμή αναπαρίσται μέσα σε ένα HashMap ως:

Χαρακτηριστικό (διάσταση)	Nominal Τιμή	$\frac{c_{i,d}}{N}$
---------------------------	--------------	---------------------



Εικόνα 13 Αναπαράσταση HashMap

όπου $c_{i,d}$ μετρητής παρουσίας της nominal τιμής i στη διάσταση d ανάμεσα σε όλα τα δεδομένα, προς τον αριθμό του συνόλου των δεδομένων. Δηλαδή δίνουμε μια πραγματική τιμή που είναι η πιθανότητα εμφάνισης της συγκεκριμένης τιμής στην συγκεκριμένη διάσταση.

Έτσι κατά την εκτέλεση του αλγορίθμου για κάθε μη αριθμητική τιμή γίνεται μια αναζήτηση στο HashMap ώστε να αντικατασταθεί με την αντίστοιχη αριθμητική της.

Η μέθοδος αυτή βρίσκεται μέσα στο αρχείο **Utils.php** και ονομάζεται **buildNominalMap()**.

6.6 Υλοποίηση συνάρτησης προσαρμογής για επιβλεπόμενη και μη επιβλεπόμενη ομαδοποίηση και της μεθόδου αποτίμησής της.

Ένα σημαντικό κομμάτι της υλοποίησής μας είναι η δημιουργία της συνάρτησης προσαρμογής για την φάση της εκπαίδευσης του αλγορίθμου. Υλοποιήσαμε τις συναρτήσεις αυτές για επιβλεπόμενη ομαδοποίηση, στο αρχείο **ClassificationFitness.java** και για μη επιβλεπόμενη ομαδοποίηση στο αρχείο **ClusteringFitness.java**.

Για την πρώτη περίπτωση η συνάρτηση προσαρμογής είναι η απόσταση κάθε εγγραφής των δεδομένων ως προς το αντίστοιχο τρέχων κέντρο του κάθε σωματιδίου της αντίστοιχης κλάσης που του έχει ήδη δοθεί. Εδώ εφαρμόζεται και η αναζήτηση στο HashMap όποτε πρέπει να υπολογιστεί μια μη αριθμητική τιμή, έτσι ώστε να γίνει η αντιστοίχισή της σε μια αριθμητική.

Στην περίπτωση της μη επιβλεπόμενης ομαδοποίησης η διαδικασία είναι λίγο πιο πολύπλοκη, επίσης δεν υπάρχει διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης και ελέγχου.

Αρχικά απομονώνουμε τα κέντρα του σωματιδίου σε μία ArrayList για να κάνουμε γρήγορα τις συγκρίσεις. Για κάθε συνδυασμό κέντρων μετράμε τις αποστάσεις και το αποτέλεσμα αυτό είναι ο παρονομαστής της συνάρτησης προσαρμογής ή αλλιώς Inter-distance, όπως έχουμε δει σε προηγούμενο κεφάλαιο. (3.1)

Μετά, κάθε εγγραφή των δεδομένων συγκρίνεται με όλα τα κέντρα των κλάσεων του σωματιδίου k ανατίθεται σε αυτό που απέχει λιγότερο. Ένας μετρητής σε κάθε κλάση μετράει το πλήθος των εγγραφών που ανήκουν σε αυτές. Έπειτα η απόσταση από την κλάση ανάθεσης διαιρείται με το πλήθος των εγγραφών της κλάσης. Αυτό είναι το Intra-distance και τελικά αυτό διαιρείται με το Inter-distance για να μας δώσει το αποτέλεσμα της συνάρτησης προσαρμογής.

6.7 Υλοποίηση κλάσης ελέγχου μετά την εκπαίδευση του αλγορίθμου κατά την οποία γίνεται ουσιαστικά μέτρηση της αποτελεσματικότητάς του.

Όπως αναφέραμε και προηγουμένως το αρχείο **TestPhase.java** πραγματοποιεί την ομαδοποίηση μετά τη φάση της εκπαίδευσης. Τα δεδομένα αποτιμούνται μέσω της ευκλείδειας απόστασης από το κέντρο της κάθε κλάσης. Έπειτα η κλάση με τη μικρότερη απόσταση είναι αυτή στην οποία γίνεται η ανάθεση. Τέλος για να προσδιοριστεί εάν η ομαδοποίηση ήταν σωστή ελέγχεται η τιμή της κλάσης που έχει προ-ανατεθεί στην συγκεκριμένη εγγραφή και εάν είχε ήδη ανατεθεί στην ίδια ομάδα τότε αυτό αυξάνει τον μετρητή των σωστών ομαδοποιήσεων κατά μια μονάδα. Το τελικό ποσοστό υπολογίζεται ως: $success \times 100 / TotalTestData$

6.8 Υλοποίηση 10-fold cross validation τεχνικής για την αποτίμηση των αποτελεσμάτων του αλγορίθμου.

Τα δεδομένα διαιρούνται σε 10 ομάδες στις 9 από τις οποίες γίνεται η εκπαίδευση και στη 1 η φάση ελέγχου. Αυτό γίνεται επαναληπτικά μέχρι που κάθε μία από τις 10 ομάδες θα έχει γίνει σύνολο ελέγχου όπως και επίσης κάθε δυνατή 9-άδα θα έχει γίνει σύνολο εκπαίδευσης. Το τελικό αποτέλεσμα είναι ο μέσος όρος των 10 επιμέρους αποτελεσμάτων.

Αρχικά τα σύνολα εκπαίδευσης και ελέγχου υπολογίζονται στο αρχείο **InitClassification.java** και η εναλλαγή των ομάδων ελέγχου και εκπαίδευσης γίνεται στο αρχείο **Dataset.java** από τη μέθοδο **foldDataset()**.

6.9 Υλοποίηση μεθόδου γραμμικής μείωσης της αδράνειας όπως περιγράφεται στο [5].

Στο αρχείο **InertiaDecrease.java** υπάρχει η μέθοδος **update()** στην οποία ορίζεται η γραμμική μείωση της αδράνειας σε κάθε επανάληψη που εκτελείται.

Στο ίδιο αρχείο υλοποιήθηκε και η δική μας πρόταση βελτιστοποίησης όπως θα περιγράψουμε στο επόμενο κεφάλαιο.

6.10 Υλοποίηση «σκορπίσματος ανέμου» όπως περιγράφεται στο [5].

Στο αρχείο **ParticleUpdateWind.java** υλοποιήσαμε την συνάρτηση «σκορπίσματος ανέμου», που λειτουργεί ως μία παράμετρος τυχαιότητας στην εξέλιξη των σωματιδίων με σκοπό την αποφυγή τοπικών βέλτιστων. Έτσι υπολογίζουμε τις 2 τυχαίες τιμές που απαιτούνται ως «άνεμος με θετική φορά» και «άνεμος με αρνητική φορά» και τις προσθέτουμε στην προηγούμενη τιμή του ανέμου. Κατά την πρώτη εκτέλεση η προηγούμενη τιμή του ανέμου είναι μια τυχαία τιμή.

6.11 Παράμετροι εκτέλεσης του PSO ομαδοποιητή

Για την εκτέλεση του αλγορίθμου πρέπει να προκαθοριστούν κάποιες βασικές παράμετροι. Η εφαρμογή μας αναπτύχθηκε με τέτοιο τρόπο ώστε ο χρήστης να μπορεί να δώσει τις τιμές των παραμέτρων πριν από την εκτέλεση. Παρόλα αυτά οι παράμετροι έχουν προκαθορισμένες τιμές ώστε να μην είναι απαραίτητο ο χρήστης να ασχοληθεί υποχρεωτικά με την ρύθμιση των τιμών.

Η εντολή εκτέλεσης του ομαδοποιητή έχει την εξής μορφή:

```
java -jar pso.jar arg1 arg2 arg3 arg4 arg5 arg6 arg7 arg8 arg9 arg10
```

Στον παρακάτω πίνακα βλέπουμε τις παραμέτρους και τις προκαθορισμένες τιμές τους:

Όνομα	Περιγραφή	Προκαθορισμένη Τιμή
Arg1	Με/χωρίς μείωση αδράνειας (true/false)	false
Arg2	Με/χωρίς βελτίωση τοπικής αναζήτησης 1 (true/false)	false
Arg3	Με/χωρίς βελτίωση τοπικής αναζήτησης 2 (true/false)	false
Arg4	Μονοπάτι και όνομα αρχείου	N/A
Arg5	Αρχική τιμή αδράνειας	0.9
Arg6	Αρχική τιμή προσωπικής παραμέτρου	0.3
Arg7	Αρχική τιμή κοινωνικής παραμέτρου	0.9
Arg8	Αριθμός επαναλήψεων	100
Arg9	Αριθμός αναδίπλωσης δεδομένων (folds)	10
Arg10	Αριθμός σωματιδίων	50

Table 4 Παράμετροι εκτέλεσης

7. ΠΡΟΤΑΣΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Στην προσπάθειά μας να εξετάσουμε μια βελτιστοποίηση στον χρόνο σύγκλισης του αλγορίθμου υλοποιήσαμε μια μέθοδο βελτίωσης της αποτελεσματικότητας του PSO με τη χρήση γραμμικής αύξησης της επιρροής της προσωπικής παραμέτρου και ταυτόχρονα γραμμικής μείωσης της επιρροής της κοινωνικής παραμέτρου, αυξάνοντας και μειώνοντας αντίστοιχα τις αντίστοιχες παραμέτρους στη συνάρτηση της ταχύτητας.

Έπειτα από έναν αριθμό επαναλήψεων το προσωρινό ολικό βέλτιστο θεωρείται ως «αξιόπιστο» δηλαδή υποψήφιο για τελικό ολικό βέλτιστο ή για αυτό που θα μας οδηγήσει στην εύρεση του τελικού που θα βρίσκεται στην περιοχή γύρω από αυτό. Έτσι κατευθύνουμε όλα τα σωματίδια που βρίσκονται στην περιοχή, προς αυτό ώστε να πραγματοποιήσουν εκεί μια αποτελεσματικότερη τοπική αναζήτηση.

Με την τεχνική αυτή αυξάνουμε την αποτελεσματικότητα του αλγόριθμου στο πέρασμα από το γενικό στάδιο αναζήτησης στο τοπικό, σταδιακά ή πιο απότομα, από κάποιο αριθμό επαναλήψεων και μετά, καθώς «εμπιστευόμαστε» περισσότερο πλέον το ολικό βέλτιστο. Το αποτέλεσμα είναι να πραγματοποιούμε μια αποτελεσματικότερη τοπική αναζήτηση πάντα γύρω από το ολικό βέλτιστο που έχει ήδη βρεθεί. Τα σωματίδια που βρίσκονται μακριά από το βέλτιστο αυτό και κοντά σε ένα άλλο τοπικό βέλτιστο θα αποπροσανατολιστούν πιθανώς αλλά τα σωματίδια που βρίσκονται ήδη γύρω από το ολικό βέλτιστο θα έχουν περισσότερες πιθανότητες να βρουν μια ακόμα καλύτερη λύση.

Δηλαδή, από τη συνάρτηση της ταχύτητας:

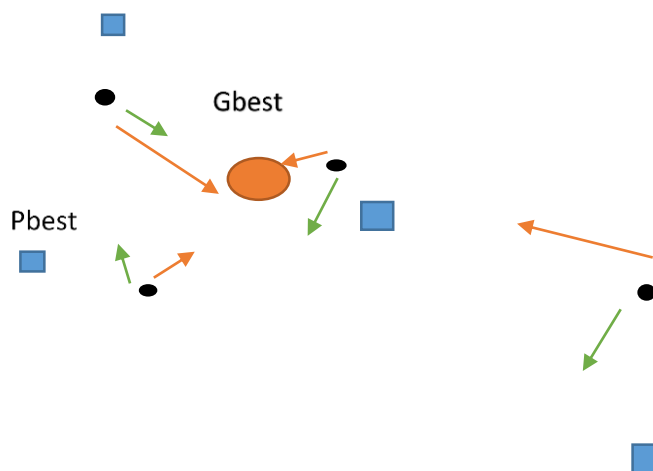
$$V_i(t+1) = W \times \vec{V}_i(t) + C_1 \times \vec{U}(0,1) \otimes (\vec{B}_i(t) - \vec{X}_i(t)) + C_2 \times \vec{U}(0,1) \otimes (\vec{B}_g(t) - \vec{X}_i(t))$$

αυξάνουμε την παράμετρο c_1 και μειώνουμε ή κρατάμε σταθερή την παράμετρο c_2 . Σε κάθε περίπτωση η προσωπική παράμετρος πρέπει να παραμένει μικρότερη από την κοινωνική, η οποία αντιπροσωπεύει την επιρροή του ολικού βέλτιστου σε όλα τα σωματίδια. Αν και αυξάνουμε την απόδοση του αλγορίθμου στο στάδιο της τοπικής αναζήτησης, τα σωματίδια πρέπει πάντα να επηρεάζονται από το ολικό βέλτιστο εφόσον το εμπιστευόμαστε ως αντιπροσωπευτικό της περιοχής βέλτιστης λύσης, σύμφωνα με τη βασική ιδέα του PSO. Αυτό γίνεται για αποφυγή αποπροσανατολισμού των σωματιδίων εξαιτίας κάποιου τοπικού βέλτιστου.

Η τεχνική αυτή βοηθάει στην γρηγορότερη σύγκλιση του αλγόριθμου και στο πέρασμα από την γενική αναζήτηση στην τοπική πιο αποτελεσματικά.

Η υλοποίηση αυτή έγινε στο αρχείο **InertiaDecrease.java** όπως αναφέραμε στο προηγούμενο κεφάλαιο αφού γίνεται παράλληλα με τη γραμμική μείωση της αδράνειας.

Η τιμή της προσωπικής παραμέτρου αυξάνεται γραμμικά μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής η οποία είναι μικρότερη ή ίση από την τιμή της κοινωνικής παραμέτρου.



Εικόνα 14 - Κατεύθυνση των σωματιδίων με και χωρίς βελτιστοποίηση προσωπικής παραμέτρου

Μία δεύτερη παραλλαγή που προτείνουμε, και υλοποιήσαμε επίσης, είναι η μετατροπή του αλγορίθμου σε PSO με γειτονιές (local best), προς το τέλος των επαναλήψεων της εκτέλεσης, κρατώντας όμως στην συνάρτηση της ταχύτητας και την επιρροή του ολικού βέλτιστου.

$$V_i(t+1) = W \times \vec{V}_i(t) + C_1 \times \vec{U}(0,1) \otimes (\vec{B}_i(t) - \vec{X}_i(t)) + C_2 \times \vec{U}(0,1) + C_1 \times \vec{U}(0,1) \otimes (\vec{B}_i(t) - \vec{X}_i(t)) + C_2 \times \vec{U}(0,1) \otimes (\vec{B}_g(t) - \vec{X}_i(t))$$

Με την τεχνική αυτή επιτυγχάνουμε ακόμα πιο αποτελεσματικά, όπως είδαμε στα πειράματα την τοπική αναζήτηση κοντά στο πιθανό ολικό βέλτιστο, ανακατευθύνοντας τα σωματίδια και προς τα γειτονικά βέλτιστα.

Ο αλγόριθμος εξελίσσεται σαν απλός PSO και η επιρροή των γειτονικών βέλτιστων, η παράμετρος c_1 είναι μηδενική για έναν αριθμό εκτελέσεων. Προς τις τελευταίες επαναλήψεις ορίζουμε τιμή στην παράμετρο και επηρεάζουμε έτσι την ταχύτητα άρα και την θέση των σωματιδίων.

Η μέθοδος αυτή δείχνει ότι η τοπική αναζήτηση είναι αποτελεσματικότερη για την εύρεση του ολικού βέλτιστου.

Και οι 2 προτάσεις βασίζονται στην πεποίθηση ότι το ολικό βέλτιστο των τελευταίων επαναλήψεων είναι πιο αξιόπιστο και αρκετά κοντά στο πραγματικό βέλτιστο. Έτσι η τοπική αναζήτηση στο σημείο αυτό είναι πολύ πιθανό να μας δώσει ένα ακόμα καλύτερο αποτέλεσμα.

8. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΩΝ

Στα πλαίσια της εργασίας, υλοποιήσαμε και προσαρμόσαμε τον αλγόριθμο PSO για classification προβλήματα ενώ πραγματοποιήσαμε επίσης κάποια πειράματα για να βγάλουμε χρήσιμα πρακτικά συμπεράσματα σχετικά με την συμπεριφορά, την απόδοση και την αποτελεσματικότητα του αλγόριθμου συγκριτικά με άλλους αλγορίθμους.

Σαν σύνολα δεδομένων για τα πειράματά μας χρησιμοποιήσαμε αρχεία ARFF τα οποία κατεβάσαμε από το διαδικτυακό τόπο του πανεπιστημίου του Waikato (<http://www.cs.waikato.ac.nz>) το οποίο διατηρεί και διαθέτει στο κοινό αρχεία τα οποία μπορούν να χρησιμοποιηθούν για ανάλυση και πειραματισμό σε διάφορα projects που αφορούν την Τεχνητή Νοημοσύνη (Μηχανική Μάθηση) και την Εξαγωγή Δεδομένων (data mining). Το πανεπιστήμιο επίσης διαθέτει λογισμικό στο οποίο υπάρχουν έτοιμες υλοποιήσεις αλγορίθμων τους οποίους χρησιμοποιήσαμε και εμείς για τις συγκρίσεις της απόδοσης του PSO με τους αλγορίθμους αυτούς. Μέχρι στιγμής δεν υπάρχει έτοιμη υλοποίηση του PSO στο λογισμικό αυτό. Γι' αυτό και υλοποιήθηκε εκ νέου ο δικός μας PSO ομαδοποιητής (classifier).

Το λογισμικό μαζί με τα αρχεία ονομάζεται **WEKA Project** και βρίσκεται στην τοποθεσία <http://www.cs.waikato.ac.nz/ml/weka>.

Τα αρχεία πάνω στα οποία πραγματοποιήθηκαν τα πειράματά μας είναι τα εξής:

- **Sonar**: Είναι ένα σύνολο δεδομένων με τις μετρήσεις οργάνων sonar και τον διαχωρισμό των μετρήσεων με βάση το αποτέλεσμα, δηλαδή εάν βρήκαν μέταλλο ή πετρώματα.
- **Waveform-5000**: Είναι ένα σύνολο δεδομένων για κατηγοριοποίηση κυμάτων βάσει της κυματομορφής τους.
- **Lymph**: Είναι ένα σύνολο δεδομένων από το ογκολογικό ινστιτούτο της Ljubljana που κατηγοριοποιεί εξετάσεις λεμφαγγειογραφίας σε διαφορετικές κλάσεις καταστάσεων του καρκίνου.

Οι αλγόριθμοι με τους οποίους συγκρίναμε τον PSO στην ομαδοποίηση των παραπάνω δεδομένων είναι οι εξής:

- **“ZeroR”**: είναι ένας αλγόριθμος που χρησιμοποιείται κυρίως σαν βάση σύγκρισης για εργασίες ομαδοποίησης. Δημιουργεί έναν πίνακα συχνότητων και επιλέγει την πιο συχνή τιμή.
- **“Naïve Bayes”**: είναι ένας πιθανοτικός αλγόριθμος ομαδοποίησης που βασίζεται στο θεώρημα του Bayes.

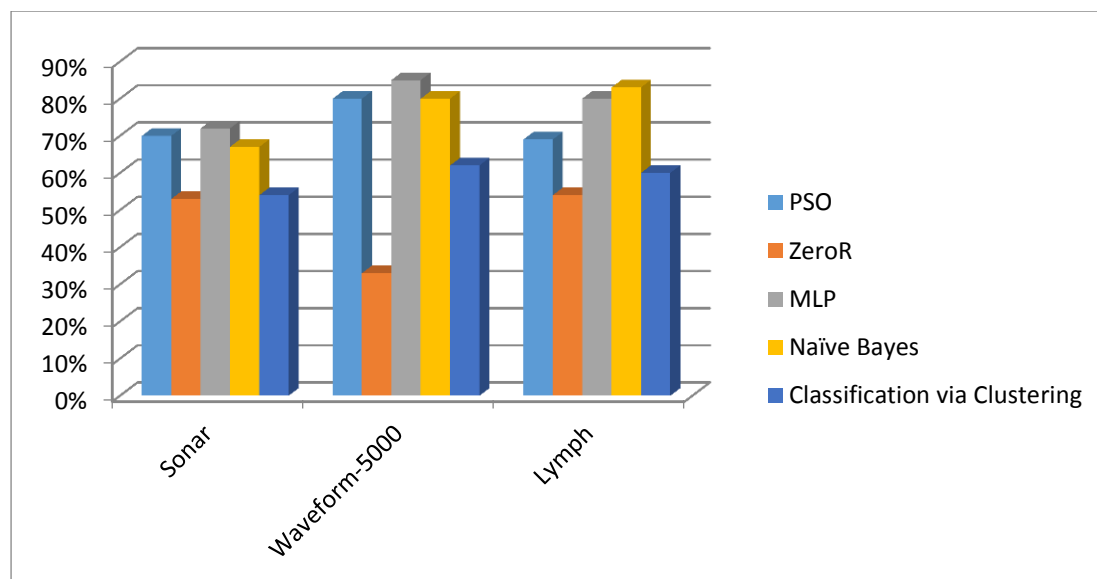
- **“Radial Basis Function Network”**: Είναι μια μορφή νευρωνικού δικτύου που χρησιμοποιεί radial basis συναρτήσεις σαν συναρτήσεις ενεργοποίησης. Η έξοδος του δικτύου είναι ένας γραμμικός συνδυασμός των εισόδων και των παραμέτρων των νευρώνων του κρυφού επιπέδου. Διατυπώθηκαν το 1988 από τους Broomhead and Lowe.
- **“Classification via Clustering”**: Είναι μια υβριδική μέθοδος επιβλεπόμενης και μη επιβλεπόμενης ομαδοποίησης. Τα δεδομένα εκπαίδευσης ομαδοποιούνται με μη επιβλεπόμενο τρόπο και δημιουργούνται οι κλάσεις δεδομένων. Οι κλάσεις αυτές αντιστοιχίζονται με τις κλάσεις της επιβλεπόμενης ομαδοποίησης. Αφού εκτελεστεί ο αλγόριθμος σε αυτά τα δεδομένα, μετά αξιολογείται με ένα σύνολο δεδομένων ελέγχου τα οποία τοποθετεί σε όποια από τις προηγούμενες κλάσεις είναι η «κοντινότερη». Μέσω της αντιστοίχισης και της ήδη προκαθορισμένης κλάσης κάθε εγγραφής των δεδομένων ελέγχου, μπορούμε να καταλάβουμε εάν η τοποθέτηση ήταν επιτυχής, άρα και να μετρήσουμε τον βαθμό ευστοχίας.

Παρακάτω θα καταγράψουμε τα αποτελέσματα των πειραμάτων μας για τις διάφορες μορφές του αλγόριθμου PSO που συγκρίναμε μεταξύ τους αλλά και με άλλους αλγόριθμους.

Πίνακας Σύγκρισης PSO με άλλους αλγόριθμους επιβλεπόμενης ομαδοποίησης σε διαφορετικά datasets. Η σύγκριση γίνεται με βάση τις πετυχημένες ομαδοποιήσεις πάνω στο test set του κάθε dataset και με τη μέθοδο του 10-fold cross validation.

Dataset/Αλγόριθμος	PSO	ZeroR	MLP	Naïve Bayes	Classification via Clustering
Sonar	70%	53%	72%	67%	54%
Waveform-5000	80%	33%	85%	80%	62%
Lymph	69%	54%	80%	83%	60%

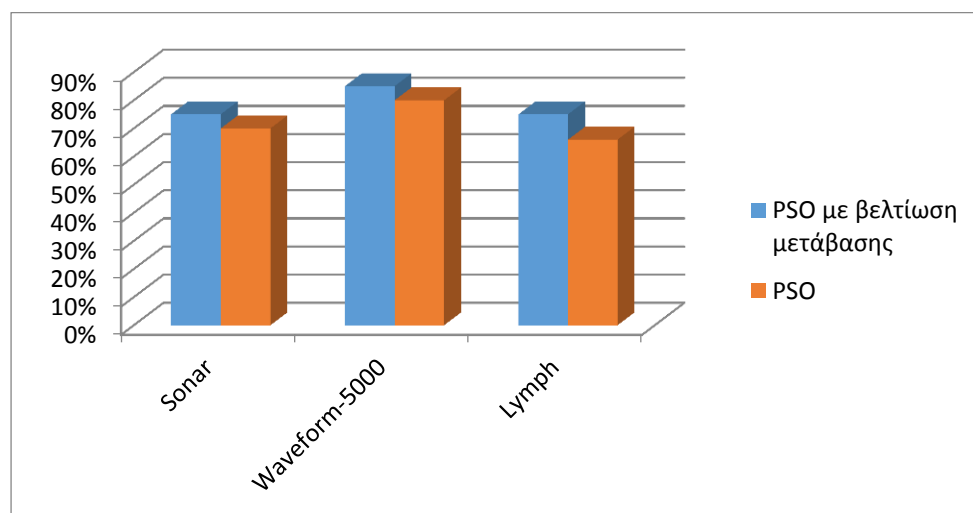
Table 5 Σύγκριση απλού PSO με άλλους αλγόριθμους ταξινόμησης/ομαδοποίησης



Πίνακας Σύγκρισης PSO με την πρώτη παραλλαγή βελτίωσης της μετάβασης. Η σύγκριση γίνεται με βάση τις πετυχημένες ομαδοποιήσεις πάνω στο test set του κάθε dataset και με τη μέθοδο του 10-fold cross validation.

Dataset/Τύπος PSO	PSO με βελτίωση μετάβασης	PSO
Sonar	75%	70%
Waveform-5000	85%	80%
Lymph	75%	69%

Table 6 Σύγκριση PSO με βελτίωση μετάβασης και απλού PSO

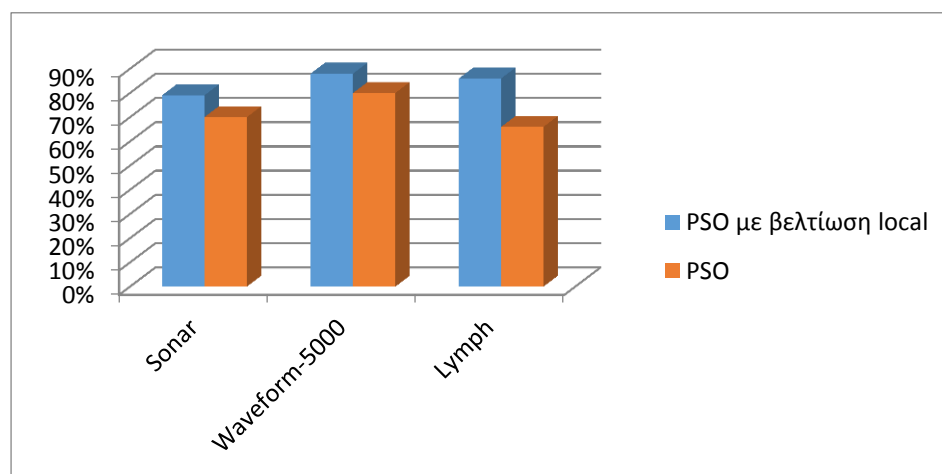


Εικόνα 15- Σύγκριση απλής βελτίωσης με τον απλό PSO

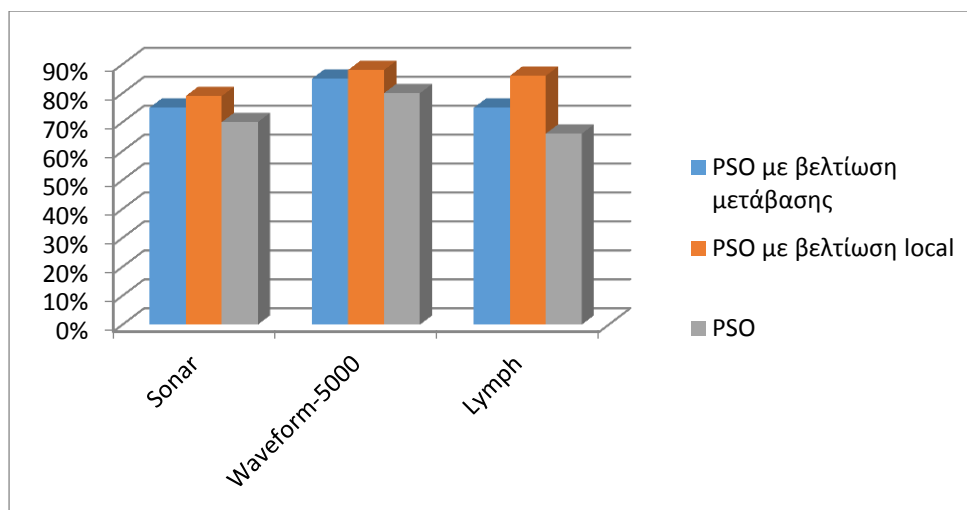
Πίνακας Σύγκρισης PSO με την δεύτερη παραλλαγή βελτίωσης της μετάβασης.

Dataset/Τύπος PSO	PSO με βελτίωση μετάβασης local	PSO
Sonar	79%	70%
Waveform-5000	88%	80%
Lymph	86%	69%

Table 7 Σύγκριση PSO με τοπική βελτίωση με τον απλό PSO



Εικόνα 16- Σύγκριση βελτίωσης local με τον απλό PSO



Εικόνα 17 - Σύγκριση απόδοσης βελτιώσεων

9. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στον επίλογο της εργασίας μας συμπεράναμε ότι η απόδοση και η αποτελεσματικότητα του PSO στην εύρεση ολικά βέλτιστης λύσης εξαρτάται σημαντικά από το πλήθος των δειγμάτων και των χαρακτηριστικών τους. Γενικά ο αλγόριθμος PSO συμπεριφέρεται καλύτερα σε πολλά και πολυδιάστατα δεδομένα σε σχέση με άλλους αλγόριθμους. Υπάρχει επίσης σημαντική βελτίωση των τελικών αποτελεσμάτων του με χρήση των παραλλαγών που αναφέραμε στο 5^ο και 7^ο κεφάλαιο.

Ένα πολύ ευαίσθητο σημείο είναι η επιλογή κατάλληλων παραμέτρων και η εξέλιξή τους κατά τη διάρκεια εκτέλεσης έτσι ώστε η μετάβαση από το γενικό στάδιο στο τοπικό να μην είναι αργή αλλά ούτε και να είναι ανεπαρκής η εκτέλεση των επιμέρους σταδίων.

Ο βασικός αλγόριθμος PSO παρατηρήσαμε ότι είναι γενικά αργός ως προς τη σύγκλιση σε μια τελική βέλτιστη λύση. Επίσης εμπεριέχει στοιχεία τυχαιότητας στις συναρτήσεις του με αποτέλεσμα τα αποτελέσματα συχνά να διαφέρουν από εκτέλεση σε εκτέλεση. Αυτό έχει ακόμα σαν αποτέλεσμα να απαιτείται μεγάλος αριθμός δεδομένων εκπαίδευσης τις περισσότερες φορές ώστε τα σωματίδια σταδιακά να συγκλίνουν προς τη σωστή κατεύθυνση.

Στο μέλλον πιστεύουμε ότι η έρευνα πρέπει να επικεντρωθεί ακριβώς στην εύρεση τεχνικών ακόμα πιο αποτελεσματικής βελτιστοποίησης του χρόνου σύγκλισης καθώς και πιο αποδοτικών μεθόδων απεγκλωβισμού των σωματιδίων από τοπικά βέλτιστα με τεχνικές αποδοτικότερης ευρείας αναζήτησης στο πρώτο μέρος των επαναλήψεων και τοπικής αναζήτησης στο τελευταίο μέρος των επαναλήψεων.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Swarm	Σμήνος
Particle	Σωματίδιο
Optimization	Βελτιστοποίηση
Clustering	Ομαδοποίηση – Συσταδοποίηση
Classification	Ομαδοποίηση - Κατηγοριοποίηση
(Un)Supervised	(Μη) Επιβλεπόμενη
Fitness function	Συνάρτηση Προσαρμογής
Objective function	Αντικειμενική Συνάρτηση
dataset	Σύνολο δεδομένων
Training set	Σύνολο εκπαίδευσης
Classifier/Clusterer	Ομαδοποιητής
Velocity Clamping	Περιορισμός Ταχύτητας

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

PSO	Particle Swarm Optimization
KNN	K nearest neighbor
NN	Neural network

ΑΝΑΦΟΡΕΣ

- [1] L. W. Yudong Zhang, «Tabu Search Particle Swarm Optimization used in Cluster Analysis,» *Journal of Science*, pp. 6-12, 2012.
- [2] S.-Y. L. Yucheng Kao, «Combining K-means and Particle Swarm Optimization for Dynamic Data Clustering Problems,» *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference*, pp. 757-761, 2009.
- [3] T. E. P. P. Xiaohui Cui, "Document Clustering using Particle Swarm Optimization," *Applied Mathematics and Computation*, pp. 337-345, 2005.
- [4] K. D. Xiaodong Li, «Comparing lbest PSO Niching algorithms Using Different Position Update Rules,» *Evolutionary Computation (CEC), 2010 IEEE Congress*, pp. 1-8, 2010.
- [5] A. S. A. N. Tiago Sousa, «Particle Swarm based Data Mining Algorithms for classification tasks,» *Journal Parallel Computing - Special issue: Parallel and nature-inspired computational paradigms and applications*, pp. 767 - 783, 2004.
- [6] Y. Tan, «Particle Swarm Optimization Algorithms Inspired by Immunity-Clonal Mechanism and Their Applications to Spam Detection,» *International Journal of Swarm Intelligence Research (IJSIR)*, pp. 64-86, 2009.
- [7] B. A. J. O. A. A. Taher Niknam, «An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering,» *Journal of Zhejiang University SCIENCE A*, pp. 512-519 , 2008.
- [8] R. W. Sabine Helwig, «Theoretical Analysis of Initial Particle Swarm Behavior,» *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pp. 889-898, 2008.
- [9] W.-M. S. W. K. B.-X. Y. Qi Shen, «A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification,» σε *Talanta*, Elsevier, 2006.
- [10] R. A. K. a. E. O. Patrick Marques Ciarelli, «Particle Swarm Optimization Applied to Parameters Learning of Probabilistic Neural Networks for Classification of Economic Activities,» σε *Particle Swarm Optimization*,

Aleksandar Lazinica.

- [11] M. B. R. P. Nabila Nouaouria, «Particle swarm classification: A survey and positioning,» σε *Pattern recognition*, Elsevier, 2012, p. 2028–2044.
- [12] M. N. Nouaouria, «A particle swarm optimization approach to mixed attribute data-set classification,» σε *Swarm Intelligence (SIS), 2011 IEEE Symposium*, 2011.
- [13] A. P. E. a. A. S. Mahamed G.H. Omran, «Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification,» σε *Pattern Analysis and Applications*, 2005, pp. 332-344.
- [14] A. E. A. S. M Omrana, «Particle Swarm Optimization Method for Image Clustering,» *International Journal of Pattern Recognition and Artificial Intelligence*, pp. 297-321, 2005.
- [15] A. S. A. E. M Omran, «Image Classification Using Particle Swarm Optimization,» *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, 2002.
- [16] Z. S. a. G. L. Lihong Xiao, «K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection,» *Intelligent Control and Automation. WCICA 2006. The Sixth World Congress*, pp. 5854 - 5858, 2006.
- [17] Y.-D. L. C.-H. Y. Li-Yeh Chuang, «An Improved Particle Swarm Optimization for Data Clustering,» *Proceedings of the international multiconference of engineers and computer scientists*, 2012.
- [18] B.-W. O. Ingyu Lee, «Role Based Particle Swarm Optimization Algorithm,» *Proceedings of International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*, 2010.
- [19] D. W. C. Hamouda K.Chantar, «Feature Subset Selection for Arabic Document Categorization using BPSO-KNN,» *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress*, pp. 546-551, 2011.
- [20] A. E. DW van der Merwe, «Data Clustering using Particle Swarm Optimization,» *Evolutionary Computation. CEC '03. The 2003 Congress*, pp. 215 - 220, 2003.
- [21] C.-H. W. Chih-Chin Lai, «Particle Swarm Optimization-Aided Feature Selection for Spam Email,» *Innovative Computing, Information and*

Control, 2007. ICICIC '07. Second International Conference, p. 165, 2007.

- [22] J. Blondin, «Particle Swarm Optimization: A Tutorial,» Department of Computer Science and Information Technology Armstrong Atlantic State University - CSCI 8100 - Computational Intelligence, 2009.
- [23] M. S. S. T. N. Bahman Bahmani Firouzi, «A new hybrid algorithm based on PSO, SA and K-means for cluster analysis,» *International Journal of Innovative Computing, Information and Control*, p. 3177–3192, 2010.
- [24] M. C. P. S. A. R. B. Jarboui, «Combinatorial particle swarm optimization (CPSO) for partitional clustering problem,» σε *Applied Mathematics and Computation*, 2007, pp. 337-345.
- [25] A. M. Alper Unler, «A discrete particle swarm optimization method for feature selection in binary classification problems,» *European Journal of Operational Research*, p. 528–539, 2010.