



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανακάλυψη πληροφορίας πλαισίου μέσω αλγορίθμου
βελτιστοποίησης σμήνους σωματιδίων**

Κωνσταντίνος Ε. Χατζηκοκόλακης

**Επιβλέποντες: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Χρήστος Αναγνωστόπουλος, Διδάκτωρ ΕΚΠΑ**

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2011

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανακάλυψη πληροφορίας πλαισίου μέσω αλγορίθμου βελτιστοποίησης σμήνους
σωματιδίων

Κωνσταντίνος Ε. Χατζηκοκολάκης

A.M.: 1115200300148

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Χρήστος Αναγνωστόπουλος, Διδάκτωρ ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Στη παρούσα πτυχιακή εργασία μελετάται η συμπεριφορά ενός συστήματος κόμβων κατά τη προσπάθειά τους να ανακαλύψουν πληροφορία πλαισίου (context). Βασικοί στόχοι είναι η ανάκτηση του context από κατά το δυνατόν περισσότερους κόμβους και η βελτιστοποίηση ως προς την κατανάλωση ενέργειας του συστήματος και τη μέση ακτίνα των κεραιών των κόμβων του συστήματος.

Στα πλαίσια της εργασίας υλοποιήθηκε ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων. Ο αλγόριθμος αυτός περιλαμβάνεται σε ένα σύνολο αλγορίθμων που βασίζονται σε κοινωνιολογικά και ψυχολογικά χαρακτηριστικά που παρατηρούνται σε σμήνη ή αγέλες. Οι αλγόριθμοι αυτοί ανήκουν στην ευρύτερη ερευνητική περιοχή της Νοημοσύνης Σμήνους.

Στον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων η κίνηση του κάθε κόμβου εξαρτάται από τις πληροφορίες που αυτός αντλεί από τους γείτονές του καθώς και από προγενέστερες μετακινήσεις του στο χώρο. Στα πλαίσια της εργασίας αναλύθηκαν τρία πιθανά σενάρια για διάφορες τοπολογίες δικτύου. Στο τελευταίο κεφάλαιο παρατίθεται ο πηγαίος κώδικας που αναπτύχθηκε κατά την εκπόνηση της εργασίας.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: π.χ. Ανακάλυψη πληροφορίας πλαισίου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: π.χ. πληροφορία πλαισίου, ανακάλυψη πλαισίου, νοημοσύνη σμήνους, βελτιστοποίηση σμήνους σωματιδίων

Στη πολυαγαπημένη και πρόωρα χαμένη μητέρα μου

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες, επίκουρο καθηγητή κ. Ευστάθιο Χατζηευθυμιάδη και τον κ. Χρήστο Αναγνωστόπουλο, για τη συνεργασία, την εμπιστοσύνη, την υπομονή και την πολύτιμη συμβολή τους στην ολοκλήρωση της.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	12
1. ΕΙΣΑΓΩΓΗ	13
2. ΠΛΗΡΟΦΟΡΙΑ ΠΛΑΙΣΙΟΥ	14
2.1 Ορισμός πληροφορίας πλαισίου	14
2.1.1 Ιστορική αναδρομή	14
2.1.2 Ορισμός	15
2.1.3 Κατηγορίες πλαισίου	15
2.2 Επίγνωση πληροφορίας πλαισίου	16
2.3 Συστήματα επίγνωσης πληροφορίας πλαισίου	17
2.3.1 Εισαγωγή	17
2.3.2 Σχεδιαστικές Αρχές	18
3. SWARM INTELLIGENCE: ΥΠΟΔΟΜΕΣ, ΠΡΟΟΠΤΙΚΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ	27
3.1 Εισαγωγή	27
3.2 Κανονική Μορφή Αλγορίθμου PSO	27
3.2.1 Κανονικό μοντέλο	27
3.2.2 Οι παράμετροι του particle swarm optimization αλγορίθμου	29
3.2.3 Τοπολογίες	30
4. ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ	34
4.1 Περιγραφή παραμέτρων προβλήματος	34
4.2 Περιγραφή Σεναρίων	34
4.2.1 Σενάριο 1 ^ο : Απόκτηση πληροφορίας πλαισίου από όλους τους κόμβους	34
4.2.2 Σενάριο 2 ^ο : Απόκτηση πληροφορίας πλαισίου από τους πλησιέστερους γείτονες	35
4.2.3 Σενάριο 3 ^ο : Απόκτηση κατά το δυνατόν ποιοτικότερης χρονικά πληροφορίας από το κάθε κόμβο	36
5. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	38
5.1 Αποτελέσματα προσομοίωσης για το 1^ο σενάριο	38

5.2	Αποτελέσματα προσομοίωσης για το 2 ^ο σενάριο	42
5.3	Αποτελέσματα προσομοίωσης για το 3 ^ο σενάριο	48
	ΣΥΜΠΕΡΑΣΜΑΤΑ	52
	ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	53
	ΣΥΝΤΜΗΣΕΙΣ - ΑΡΚΤΙΚΟΛΕΞΑ - ΑΚΡΩΝΥΜΙΑ.....	55
	ΠΑΡΑΡΤΗΜΑ Ι	56
	ΑΝΑΦΟΡΕΣ.....	72

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Διάταξη κόμβων	34
Σχήμα 2: Διάδοση πληροφορίας σε δίκτυο με 100 κόμβους	39
Σχήμα 3: Διάδοση πληροφορίας σε δίκτυο με 900 κόμβους	39
Σχήμα 4: Απώλεια ενέργειας συστήματος για δίκτυο με 100 κόμβους	40
Σχήμα 5: Απώλεια ενέργειας συστήματος για δίκτυο με 900 κόμβους	40
Σχήμα 6: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους	41
Σχήμα 7: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους	41
Σχήμα 8: Απώλεια ενέργειας συστήματος για δίκτυο με 100 κόμβους	42
Σχήμα 9: Απώλεια ενέργειας συστήματος για δίκτυο με 900 κόμβους	42
Σχήμα 10: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους	43
Σχήμα 11: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους	43
Σχήμα 12: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους	44
Σχήμα 13: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους	44
Σχήμα 14: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους	45
Σχήμα 15: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους	45
Σχήμα 16: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους	46
Σχήμα 17: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους	46
Σχήμα 18: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους	47
Σχήμα 19: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους	47
Σχήμα 20: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους	48
Σχήμα 21: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους	48
Σχήμα 22: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους	49
Σχήμα 23: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους	49
Σχήμα 24: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους	50
Σχήμα 25: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους	50
Σχήμα 26: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους	51

Σχήμα 27: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους51

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Μία μονάδα για εντοπισμό θέσης τύπου Bat.....	23
Εικόνα 2: Εγκατάσταση ενός αισθητήρα.....	23
Εικόνα 3: Η βασική δομή του PSO	30
Εικόνα 4: Γραφική αναπαράσταση ενός (α)gbest swarm (β) lbest swarm.....	31
Εικόνα 5: Παραδείγματα (α) γεωγραφικής γειτονιάς (β) κοινωνικής γειτονιάς	32
Εικόνα 6: Παραδείγματα τοπολογιών	33

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Επίπεδα λειτουργικότητας για συστήματα επίγνωσης πλαισίου.....	20
Πίνακας 2: Συχνά χρησιμοποιούμενοι τύποι αισθητήρων.....	21
Πίνακας 3: Χρήση κλιμακούμενων ονομάτων για μεθόδους	25
Πίνακας 4: Ψευδοκώδικας για το κανονικό μοντέλο του αλγορίθμου PSO.....	29
Πίνακας 5 : Ψευδοκώδικας περίπτωσης απόκτησης πληροφορίας πλαισίου από όλους τους κόμβους χωρίς περαιτέρω κριτήρια	35
Πίνακας 6 : Ψευδοκώδικας περίπτωσης απόκτησης πληροφορίας πλαισίου από τον πλησιέστερο γείτονα	36
Πίνακας 7 : Ψευδοκώδικας περίπτωσης απόκτησης κατά το δυνατόν ποιοτικότερης χρονικά πληροφορίας πλαισίου	37

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια του Προπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών το ακαδημαϊκό έτος 2007-2008. Αντικείμενο μελέτης της είναι η ανακάλυψη και η επίγνωση πληροφορίας πλαισίου σε συστήματα που μοντελοποιούνται με βάση τον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων.

1. ΕΙΣΑΓΩΓΗ

Ένα βασικό χαρακτηριστικό της επικοινωνίας και της αλληλεπίδρασης ανθρώπου με άνθρωπο είναι η μεταβίβαση ιδεών και ερεθισμάτων με βάση πληροφορίες του περιβάλλοντος χώρου με έμμεσο τρόπο (κινήσεις σώματος, χειριών και προσώπου, μορφασμοί προσώπου κ.α.). Δυστυχώς, η ικανότητα μεταβίβασης αυτού του είδους πληροφορίας δεν γίνεται εξίσου καλά κατά την αλληλεπίδραση ανθρώπου με υπολογιστή. Οι υπολογιστές δεν είναι σε θέση να εκμεταλλευτούν πλήρως τις συνθήκες υπό τις οποίες ο χρήστης τους χρησιμοποιεί. Βελτώνοντας τη πρόσβαση του υπολογιστή σε τέτοιου είδους πληροφορίες, εμπλουτίζεται η επικοινωνία κατά την αλληλεπίδραση ανθρώπου-υπολογιστή, με αποτέλεσμα να είναι εφικτή η παραγωγή όσο το δυνατόν χρησιμότερων υπηρεσιών για το χρήστη.

Επομένως, στόχος της υπολογιστικής επίγνωσης πλαισίου θα πρέπει να είναι να κάνει ευκολότερη την αλληλεπίδραση υπολογιστή-ανθρώπου. Η ανάκτηση πληροφορίας πλαισίου θα πρέπει να γίνεται από τη συσκευή μέσω αισθητήρων και όχι να επαφίεται στο χρήστη, καθώς μια τέτοια διαδικασία μπορεί να αποδειχθεί δύσκολη και κουραστική για αυτόν, ενώ είναι πολύ πιθανό να μη είναι σε θέση να αναγνωρίσει ποιές πληροφορίες του περιβάλλοντός του είναι χρήσιμες για την εφαρμογή.

Η Νοημοσύνη Σμήνους (Swarm Intelligence, SI) είναι ένα σύνολο από καινοτόμους, κατανοημένους και ευφυείς αλγορίθμους για την επίλυση προβλημάτων βελτιστοποίησης. Ένας από αυτούς είναι ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization, PSO) ο οποίος υιοθετεί συνεργατικές συμπεριφορές των κόμβων του συστήματος, όπως αυτές έχουν παρατηρηθεί σε φυσικές ομάδες, όπως σμήνη εντόμων και σμήνη πτηνών.

Στη παρούσα πτυχιακή αναλύονται αρχικά οι όροι πληροφορία πλαισίου (context) και επίγνωση πληροφορίας πλαισίου (context awareness), περιγράφονται οι κατηγορίες πλαισίου καθώς επίσης και τα συστήματα επίγνωσης πληροφορίας πλαισίου (κεφάλαιο 2). Ακολούθως, στο κεφάλαιο 3, περιγράφεται διεξοδικά η Νοημοσύνη Σμήνους και η κανονική μορφή του αλγορίθμου Particle Swarm Optimization. Έπειτα, στο κεφάλαιο 4, αναλύονται οι λεπτομέρειες υλοποίησης του αλγορίθμου και τα σενάρια τα οποία μελετήθηκαν. Τέλος, στο κεφάλαιο 5 παρουσιάζονται τα πειραματικά αποτελέσματα της προσομοίωσης των σεναρίων που αναλύονται στο κεφάλαιο 4 ως προς διάφορες μετρικές, όπως η διάδοση της πληροφορίας πλαισίου στους κόμβους του δικτύου και η μέση κατανάλωση ενέργειας του συστήματος.

2. ΠΛΗΡΟΦΟΡΙΑ ΠΛΑΙΣΙΟΥ

Η χρήση του context είναι απαραίτητη για διαδραστικές εφαρμογές. Πιο συγκεκριμένα, είναι αναγκαία σε εφαρμογές όπου το πλαίσιο του χρήστη αλλάζει γρήγορα, όπως είναι οι εφαρμογές κινητών συσκευών (PDA's και κινητών τηλεφώνων), καθώς επίσης και στο τομέα της διάχυτης υπολογιστικής (ubiquitous computing). Στην ενότητα αυτή αποσαφηνίζονται οι ακόλουθες έννοιες:

- Επίγνωση Πληροφορίας Πλαισίου (context awareness)
- Εφαρμογές επίγνωσης πληροφορίας πλαισίου (context-aware applications)
- Συστήματα επίγνωσης πληροφορίας πλαισίου (context-aware systems)

με σκοπό τη καλύτερη κατανόηση των ευρύτερων περιοχών όπου βρίσκεται εφαρμογή ή επίγνωση πληροφορίας πλαισίου και τη διευκόλυνση ως προς τη κατασκευή εφαρμογών επίγνωσης πλαισίου.

2.1 Ορισμός πληροφορίας πλαισίου

2.1.1 Ιστορική αναδρομή

Τα τελευταία χρόνια πολλοί ερευνητές έχουν προσπαθήσει να ορίσουν τι είναι πλαίσιο. Ενώ οι περισσότεροι άνθρωποι μπορούν να αντιληφθούν τι είναι πλαίσιο, βρίσκουν δύσκολο να επεξηγήσουν τον όρο. Οι ορισμοί που έχουν δοθεί στο παρελθόν χωρίζονται σε δύο κατηγορίες:

- 1) απαρίθμηση παραδειγμάτων και
- 2) επιλογή συνώνυμων λέξεων ή εκφράσεων

Οι Schilit και Theimer, που ήταν οι πρώτοι οι οποίοι εισήγαγαν τον όρο επίγνωση πλαισίου [6], θεωρούν ότι το πλαίσιο χρήστη απαρτίζεται από την θέση του, ένα σύνολο από αναγνωριστικά άλλων ανθρώπων και αντικειμένων (identities) και τις αλλαγές οι οποίες συμβαίνουν σε αυτά. Παρόμοιοι ορισμοί έχουν δοθεί από τον Brown [8] και τον Ryan [9] οι οποίοι απλά διευρύνουν τον όρο προσθέτοντας κι άλλα χαρακτηριστικά που εν δυνάμει είναι σχετικά με το χρήστη, όπως η ώρα της ημέρας, εποχή, θερμοκρασία κ.α. κατά την ανάλυση του Brown, ή γενικότερα το περιβάλλον και ο χρόνος στην ανάλυση του Ryan. Αυτοί οι ορισμοί που καθορίζουν το πλαίσιο με παραδείγματα βρίσκουν δύσκολα εφαρμογή, καθώς όταν είναι σημαντικό να καθοριστεί αν ένα είδος πληροφορίας είναι ή όχι πλαίσιο, δεν είναι σαφές πως μπορούμε να χρησιμοποιήσουμε τον ορισμό για να επιλύσουμε το δίλημμα.

Άλλοι ορισμοί απλά παρέχουν συνώνυμα του πλαισίου όπως περιβάλλον ή κατάσταση. Και μάλιστα κάποιοι αντιλαμβάνονται το περιβάλλον ως περιβάλλον του χρήστη, ενώ άλλοι ως περιβάλλον της εφαρμογής. Ο Brown [13] όρισε το πλαίσιο ως τα στοιχεία του περιβάλλοντος του χρήστη για τα οποία έχει γνώση ο υπολογιστής του. Οι Franklin και FlaschBart [14] θεωρούν πλαίσιο τη κατάσταση του χρήστη. Ο Ward [15] αντιλαμβάνεται το πλαίσιο ως τη κατάσταση αυτών που περιτριγυρίζουν την εφαρμογή και ο Rodden [16] ως τη τοποθεσία της εφαρμογής. Ο Hull [17] συμπεριέλαβε όλο το περιβάλλον καθορίζοντας ως πλαίσιο την όψη της τρέχουσας κατάστασης. Όπως και στη περίπτωση των ορισμών με παραδείγματα, οι ορισμοί που επιλέγουν συνώνυμα του πλαισίου δύσκολα βρίσκουν εφαρμογή.

Οι ορισμοί των Scilit et al. [10], Dey et al. [11] και Pascoe [12] προσεγγίζουν καλύτερα την έννοια του πλαισίου καθώς δεν την περιορίζουν σε μερικά συνώνυμα ή σε παραδείγματα, αλλά δίνουν μια γενικότερη περιγραφή των σημαντικότερων πλευρών της έννοιας. Οι Scilit et al. υποστηρίζουν ότι οι σημαντικότερες όψεις του πλαισίου είναι: το που είσαι, με ποιόν είσαι μαζί και τι πόροι βρίσκονται κοντά και ορίζουν ως πλαίσιο το διαρκώς μεταβαλλόμενο περιβάλλον εκτέλεσης. Ενώ με την έννοια περιβάλλον εννοούν:

- 1) τους διαθέσιμους υπολογιστές, συσκευές και δίκτυο (computing environment),
- 2) τη τοποθεσία του χρήστη, τους ανθρώπους που είναι κοντά του και τη κοινωνική κατάσταση (user environment) και
- 3) το θόρυβο και το φως (physical environment)

Οι Dey et al. ορίζουν το πλαίσιο ως τη φυσική, κοινωνική, συναισθηματική και πληροφοριακή κατάσταση του χρήστη. Τέλος, ο Pascoe ορίζει το πλαίσιο ως το υποσύνολο των φυσικών και εννοιολογικών καταστάσεων που ενδιαφέρουν μία συγκεκριμένη οντότητα [1][3].

2.1.2 Ορισμός

Αυτοί οι ορισμοί που δόθηκαν είναι πολύ συγκεκριμένοι με αποτέλεσμα να είναι δύσκολο να εφαρμοστούν. Το τι θεωρείται πλαίσιο έχει να κάνει με τη συνολική κατάσταση που σχετίζεται με μία εφαρμογή και το σύνολο των χρηστών της. Δεν γίνεται να απαριθμήσουμε ποιά θέματα όλων των καταστάσεων είναι σημαντικά, καθώς αυτά αλλάζουν ανάλογα με τη κατάσταση στην οποία βρίσκεται ο χρήστης. Για παράδειγμα το φυσικό περιβάλλον όπως το χαρακτηρίζουν οι Scilit et al. σε κάποιες περιπτώσεις μπορεί να είναι σημαντικό για την εφαρμογή, ενώ σε άλλες μπορεί να μην έχει καμία χρησιμότητα. Συνοψίζοντας τα παραπάνω καταλήγουμε στον ακόλουθο ορισμό:

«Πληροφορία πλαισίου είναι οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει τη κατάσταση μίας οντότητας. Οντότητα μπορεί να είναι ένας άνθρωπος, ή ένα μέρος, ή ένα αντικείμενο το οποίο θεωρείται σχετικό στην αλληλεπίδραση ανάμεσα σε έναν χρήστη και μία εφαρμογή, συμπεριλαμβανομένων τον χρήστη και την εφαρμογή».

Ο ορισμός αυτός επιτρέπει στο πλαίσιο είτε να υπονοείται, ή να δηλώνεται ρητά από το χρήστη [1][3].

2.1.3 Κατηγορίες πλαισίου

Η τοποθεσία, η ταυτότητα, ο χρόνος και η δραστηριότητα είναι οι κυριότεροι τύποι πλαισίου για το χαρακτηρισμό της κατάστασης μίας οντότητας. Αυτοί οι τύποι πλαισίου δεν απαντούν μόνο τις ερωτήσεις ποιός, τι, που και πότε, αλλά λειτουργούν σαν ενδείξεις και για άλλες πηγές πληροφορίας πλαισίου. Για παράδειγμα, δεδομένου ενός ανθρώπου, μπορούμε να αποκτήσουμε πολλές σχετικές με αυτόν πληροφορίες, όπως τη διεύθυνσή του, το τηλέφωνό του, την διεύθυνση ηλεκτρονικού ταχυδρομείου του, πιθανές σχέσεις με άλλους ανθρώπους από το περιβάλλον του κλπ. Γνωρίζοντας τη τοποθεσία μίας οντότητας μπορούμε να αποφανθούμε τι άλλα αντικείμενα ή άνθρωποι είναι κοντά στην οντότητα αυτή και τι είδους δραστηριότητα λαμβάνει χώρα. Όπως φαίνεται και από αυτά τα παραδείγματα, οι κυριότεροι τύποι πλαισίου για μία οντότητα

μπορούν αποτελούν ενδείξεις τόσο για άλλες πληροφορίες σχετικές με την οντότητα, όσο και για πληροφορία πλαισίου σχετική με άλλες οντότητες.

2.2 Επίγνωση πληροφορίας πλαισίου

Η έννοια του context awareness ξεκίνησε από την ερευνητική περιοχή ubiquitous computing ή αλλιώς pervasive computing η οποία προσπάθησε να ασχοληθεί με τη σύνδεση των μεταβολών στο περιβάλλον με τα συστήματα πληροφορικής, τα οποία διαφορετικά θεωρούνται στατικά. Αν και ξεκίνησε ως ένας όρος επιστήμη των υπολογιστών, έχει επίσης εφαρμοστεί στη θεωρία των επιχειρήσεων όσον αφορά θέματα διαχείρισης διαδικασιών των επιχειρήσεων.

Στην επιστήμη των υπολογιστών στηρίζεται στην ιδέα ότι οι υπολογιστές μπορούν τόσο να αντιλαμβάνονται, όσο και να αντιδρούν με βάση το περιβάλλον τους. Οι συσκευές μπορούν να έχουν πληροφορίες για τις συνθήκες, υπό τις οποίες μπορούν να λειτουργούν και να βασίζονται σε κανόνες (ή σε ένα ευφυές ερέθισμα στη περίπτωση αισθητήρων) και να αντιδρούν αναλόγως. Ο όρος context awareness στην διάχυτη υπολογιστική (pervasive computing) εισήχθη από τον Schilit [7][10]. Οι συσκευές που έχουν αυτή την ικανότητα ενδέχεται επίσης να προσπαθήσουν να κάνουν υποθέσεις σχετικά με την τρέχουσα κατάσταση του χρήστη.

Ενώ η κοινότητα της επιστήμης των υπολογιστών αντιλήφθηκε αρχικά την έννοια του context ως θέμα εντοπισμού της θέσης ενός χρήστη, όπως αναφέρει ο Dey στο [3], τα τελευταία χρόνια αυτή η ιδέα έχει μελετηθεί όχι απλά ως κατάσταση, αλλά ως μέρος μίας διαδικασίας με την οποία σχετίζονται και οι χρήστες, με αποτέλεσμα να έχουν προταθεί εξελιγμένα και γενικότερα μοντέλα πλαισίου για την υποστήριξη συγκεκριμένων context-aware applications που τα χρησιμοποιούν για :

- 1.τη προσαρμογή διεπαφών
- 2.τη προσαρμογή του σύνολο των δεδομένων που σχετίζονται με την εφαρμογή
- 3.την αύξηση της ακρίβειας της ανάκτησης πληροφοριών
- 4.την ανακάλυψη υπηρεσιών
- 5.την πληρότητα της αλληλεπίδρασης με το χρήστη
- 6.τη κατασκευή έξυπνων περιβαλλόντων

Για παράδειγμα ένα κινητό τηλέφωνο το οποίο έχει επίγνωση πληροφορίας πλαισίου, θα δύναται να γνωρίζει ότι αυτή τη στιγμή βρίσκεται στην αίθουσα συνεδριάσεων, καθώς και ότι ο χρήστης έχει καθίσει. Έπειτα, ακολουθώντας τους κανόνες με τους οποίους λειτουργεί, η συσκευή μπορεί να καταλήξει στο συμπέρασμα ότι ο χρήστης βρίσκεται αυτή τη στιγμή σε μία συνάντηση και να απορρίπτει κάθε ασήμαντη κλήση.

Τα context-aware systems ασχολούνται με την ανακάλυψη πληροφορίας πλαισίου (π.χ. χρησιμοποιώντας αισθητήρες για να αντιληφθούν μία κατάσταση), με την άντληση και τη κατανόηση πλαισίου (π.χ. αντιστοίχιση ενός ερεθίσματος που αντιλήφθηκε ο αισθητήρας σε πληροφορία πλαισίου) και με την εφαρμογή συμπεριφοράς με βάση την αναγνωρισμένη πληροφορία πλαισίου (για παράδειγμα ενεργοποίηση ενεργειών με βάση το πλαίσιο) [19]. Καθώς η δραστηριότητα και η τοποθεσία του χρήστη αποτελούν παραμέτρους ζωικής σημασίας για πολλές εφαρμογές, η επίγνωση πληροφορίας πλαισίου έχει εστιάσει περισσότερο σε ερευνητικές περιοχές επίγνωσης τοποθεσίας (location awareness) και αναγνώρισης δραστηριότητας (activity recognition) [6].

Η επίγνωση πληροφορίας πλαισίου θεωρείται μία ενεργή τεχνολογία για τα συστήματα διάχυτης υπολογιστικής (ubiquitous computing systems). Χρησιμοποιείται για το σχεδιασμό καινοτόμων και πρωτοποριακών διεπαφών χρήστη και συχνά

χρησιμοποιείται ως κομμάτι της πανταχού παρούσας υπολογιστικής. Έχει αρχίσει να γίνεται αισθητή η ανάγκη της και στο διαδίκτυο με την έλευση των υβριδικών μηχανών αναζήτησης. Ο ανθρώπινος παράγοντας και το φυσικό περιβάλλον θεωρούνται δύο σημαντικές πτυχές στην επιστήμη των υπολογιστών όπως καθορίζουν οι Beigl και Gellersen [18].

Οι ανθρώπινοι παράγοντες που σχετίζονται με το πλαίσιο διαρθρώνονται σε τρεις κατηγορίες :

1. Πληροφορίες για το χρήστη
2. Κοινωνικό περιβάλλον του χρήστη
3. Καθήκοντα/Εργασίες χρήστη

Για παράδειγμα, η επίγνωση των συνθηκών του χρήστη, της συναισθηματικής του κατάστασης, των βιοψυχολογικών του συνθηκών κ.α. αποτελούν πληροφορίες σχετικές με τον ίδιο το χρήστη. Ο εντοπισμός τοποθεσίας άλλων χρηστών, η κοινωνική αλληλεπίδραση, η δυναμική της ομάδας κ.α. συγκαταλέγονται σε πληροφορίες σχετικές με το κοινωνικό περιβάλλον του χρήστη, ενώ στοιχεία όπως η αυθόρμητη δραστηριότητα, οι προγραμματισμένες εργασίες και οι γενικότεροι επαγγελματικοί και προσωπικοί στόχοι του χρήστη εντάσσονται στη κατηγορία των καθηκόντων/εργασιών του χρήστη.

Ομοίως, το πλαίσιο που σχετίζεται με το φυσικό περιβάλλον κατηγοριοποιείται σε τρεις κατηγορίες:

1. Τοποθεσία
2. Υποδομή
3. Φυσικές συνθήκες

Για παράδειγμα η απόλυτη θέση του χρήστη, η σχετική θέση σε σχέση με κάποιον άλλο χρήστη ή κάποιο αντικείμενο-χώρο αποτελούν ένα πλαίσιο πληροφοριών σχετικών με την τοποθεσία του χρήστη. Ακόμα, πληροφορίες όπως πλησιέστεροι πόροι για υπολογισμούς, επικοινωνία και βελτίωση της απόδοσης αφορούν ένα πλαίσιο πληροφοριών σχετικό με την υποδομή του συστήματος. Τέλος, ο θόρυβος, το φως, η πίεση κ.α. αποτελούν τυπικές φυσικές συνθήκες με τις οποίες σχετίζεται ένα πλαίσιο πληροφοριών [6].

2.3 Συστήματα επίγνωσης πληροφορίας πλαισίου

2.3.1 Εισαγωγή

Τα συστήματα επίγνωσης πληροφορίας πλαισίου προσφέρουν νέες ευκαιρίες τόσο για τους προγραμματιστές εφαρμογών, όσο και για τους τελικούς χρήστες, με τη συλλογή δεδομένων πλαισίου και την προσαρμογή της συμπεριφοράς των συστημάτων αναλόγως. Ιδιαίτερα σε συνδυασμό με τις κινητές συσκευές οι μηχανισμοί αυτοί έχουν μεγάλη αξία και χρησιμοποιούνται παρά πολύ για την αύξηση της χρηστικότητας των συσκευών. Είναι σε θέση να προσαρμόσουν τις λειτουργίες τους στο τρέχον πλαίσιο χωρίς ρητή παρέμβαση του χρήστη και έτσι, αποσκοπούν στην αύξηση της χρηστικότητας και της αποτελεσματικότητας, λαμβάνοντας υπόψη τους το περιβαλλοντικό πλαίσιο. Ιδιαίτερα όταν πρόκειται για τη χρήση κινητών συσκευών, είναι επιθυμητό τα προγράμματα και οι υπηρεσίες να αντιδρούν με συγκεκριμένο τρόπο

ανάλογα με την τρέχουσα θέση, τη τρέχουσα χρονική στιγμή και άλλα χαρακτηριστικά του περιβάλλοντος και να προσαρμόζουν τη συμπεριφορά τους σύμφωνα με τις μεταβαλλόμενες συνθήκες καθώς το πλαίσιο στο οποίο βρίσκεται ο χρήστης μπορεί να αλλάζει γρήγορα. Οι πληροφορίες πλαισίου που απαιτούνται μπορούν να ανακαλυφθούν με διάφορους τρόπους, όπως με την εφαρμογή αισθητήρων, μέσω πληροφοριών δικτύου, λόγω της κατάστασης των συσκευών, με περιήγηση στα προφίλ των χρηστών, ή με τη χρήση άλλων πηγών. Η ιστορία των συστημάτων επίγνωσης πλαισίου ξεκίνησε όταν οι Ward et al. (1992) παρουσίασαν το Active Badge Location System το οποίο θεωρείται ως ένα από τα πρώτα συστήματα επίγνωσης πλαισίου. Η τεχνολογία υπερύθρων στην οποία βασίζεται το σύστημα είναι σε θέση να καθορίσει τη τρέχουσα θέση ενός χρήστη, η οποία χρησιμοποιήθηκε για να διαβιβάσει τηλεφωνικές κλήσεις σε μια συσκευή τηλεφώνου κοντά στο χρήστη.

Αν και οι πληροφορίες τοποθεσίας είναι μακράν το πιο συχνά χρησιμοποιούμενο χαρακτηριστικό της πληροφορίας πλαισίου, τα τελευταία χρόνια γίνονται όλο και περισσότερες προσπάθειες να χρησιμοποιηθούν άλλες πληροφορίες πλαισίου. Ως εκ τούτου, όπως αναλύθηκε και σε προηγούμενη ενότητα, είναι ένα δύσκολο έργο ο καθορισμός της έννοιας του «πλαίσιο» και πολλοί ερευνητές προσπάθησαν να βρουν το δικό τους ορισμό για το τι περιλαμβάνει η έννοια στην πραγματικότητα.

Ένας δημοφιλής τρόπος για να κατηγοριοποιηθούν οι περιπτώσεις πληροφορίας πλαισίου είναι η διάκριση των διαφορετικών διαστάσεων του πλαισίου. Οι Prekor και Burnett (2003) και ο Gustavsen (2002) κατονομάζουν αυτές τις διαστάσεις ως εξωτερική και εσωτερική, ενώ οι Hofer et al. (2002) αναφέρονται σε φυσικό και λογικό πλαίσιο. Η εξωτερική (φυσική) διάσταση αναφέρεται σε πληροφορία πλαισίου που μπορεί να μετρηθεί με αισθητήρες υλικού (hardware sensors), δηλαδή, τη θέση, φως, ήχο, κίνηση, επαφή, θερμοκρασία ή τη πίεση του αέρα, ενώ η εσωτερική (λογική) διάσταση ως επί το πλείστον καθορίζεται από τον χρήστη ή γίνεται αντιληπτή από την παρακολούθηση των αλληλεπιδράσεων των χρηστών, για παράδειγμα, τους στόχους, τα καθήκοντα, το πλαίσιο εργασίας και επιχειρηματικές διαδικασίες, του χρήστη καθώς και τη συναισθηματική του κατάσταση. Τα περισσότερα συστήματα επίγνωσης πλαισίου χρησιμοποιούν εξωτερικούς παράγοντες, αν αυτοί μπορούν να παρέχουν χρήσιμα στοιχεία, όπως πληροφορίες τοποθεσίας. Επιπλέον, τα εξωτερικά χαρακτηριστικά είναι εύκολο να γίνουν αντιληπτά, χρησιμοποιώντας τεχνολογίες ανίχνευσης.

Ακολούθως, παρουσιάζονται οι κοινές αρχές της αρχιτεκτονικής δημοφιλών συστημάτων επίγνωσης πλαισίου και παράγεται ένα πολυεπίπεδο εννοιολογικό πλαίσιο του σχεδιασμού για την ανάλυση των διαφορετικών στοιχείων που είναι κοινά στις περισσότερες αρχιτεκτονικές επίγνωσης πλαισίου [2].

2.3.2 Σχεδιαστικές Αρχές

Σε αυτή την υποενότητα αναλύονται οι σχεδιαστικές αρχές των συστημάτων επίγνωσης πληροφορίας πλαισίου, ως προς την αρχιτεκτονική των συστημάτων αυτών και ως προς τα διαφορετικά μοντέλα πληροφορίας πλαισίου.

2.3.2.1 Αρχιτεκτονική

Συστήματα επίγνωσης πληροφορίας πλαισίου μπορούν να υλοποιηθούν με πολλούς τρόπους. Η προσέγγιση εξαρτάται από τις ειδικές απαιτήσεις και προϋποθέσεις, όπως η θέση των αισθητήρων (μπορούν είτε να βρίσκονται τοπικά, ή απομακρυσμένα), το πλήθος των πιθανών χρηστών (ένας χρήστης, ή πολλοί), οι διαθέσιμοι πόροι των συσκευών (high-end-υπολογιστές, ή μικρές φορητές συσκευές) ή την εγκατάσταση περαιτέρω επέκτασης του συστήματος. Επιπλέον, η μέθοδος απόκτησης των

δεδομένων πλαισίου είναι πολύ σημαντική κατά τη σχεδίαση του συστήματος, διότι προκαθορίζει το αρχιτεκτονικό ύψος της συστήματος, σε κάποιο βαθμό. Για την ανακάλυψη πληροφορίας πλαισίου μπορεί να ακολουθηθεί οποιαδήποτε από τις εξής προσεγγίσεις:

1. Μέθοδος άμεσης πρόσβασης σε αισθητήρα (Direct Sensor Access): Η προσέγγιση αυτή χρησιμοποιείται συχνά σε συσκευές με ενσωματωμένους αισθητήρες. Το λογισμικό του πελάτη συγκεντρώνει τις επιθυμητές πληροφορίες απευθείας από αυτούς τους αισθητήρες, δηλαδή, δεν υπάρχει κανένα επιπλέον επίπεδο για την απόκτηση και την επεξεργασία των δεδομένων αισθητήρα. Οι οδηγοί εγκατάστασης (drivers) των αισθητήρων είναι ενοποιημένοι μέσα στην εφαρμογή, με αποτέλεσμα αυτή η μέθοδος να μπορεί να χρησιμοποιηθεί μόνο σε σπάνιες περιπτώσεις. Ως εκ τούτου, δεν είναι κατάλληλη για καταναμημένα συστήματα, λόγω της φύσης της που απαιτεί άμεση πρόσβαση και που στερείται ενός στοιχείου ικανού να διαχειριστεί πολλαπλές ταυτόχρονες προσβάσεις σε αισθητήρες.
2. Υποδομή ενδιάμεσου (Middleware infrastructure): Μοντέρνες προσεγγίσεις σχεδίασης λογισμικού χρησιμοποιούν μεθόδους ενθυλάκωσης για να διαχωριστούν, π.χ., το επιχειρησιακό τομέα, το λογικό κομμάτι και τις γραφικές διεπαφές χρήστη. Η middleware προσέγγιση εισάγει μια πολυεπίπεδη αρχιτεκτονική στα συστήματα επίγνωσης πλαισίου με σκοπό την απόκρυψη χαμηλού επιπέδου λεπτομερειών αίσθησης. Σε σύγκριση με τη προσέγγιση άμεσης πρόσβασης στον αισθητήρα, σε αυτήν την τεχνική διευκολύνεται ή επεκτασιμότητα δεδομένου ότι ο κώδικας του πελάτη δεν τροποποιείται πλέον και απλοποιεί την επαναχρησιμοποίηση του εξαρτώμενου από το υλικό κώδικα του αισθητήρα λόγω της αυστηρής ενθυλάκωσης.
3. Εξυπηρετητής πληροφορίας πλαισίου (Context server): Το επόμενο λογικό βήμα είναι να επιτραπεί πολλαπλή πρόσβαση των πελατών σε απομακρυσμένες πηγές δεδομένων. Αυτή η καταναμημένη προσέγγιση διευρύνει τη στηριζόμενη σε middleware infrastructure προσέγγιση με την εισαγωγή ενός απομακρυσμένου συστατικού υπεύθυνου για τη διαχείριση πρόσβασης. Η συλλογή των δεδομένων αισθητήρα μεταφέρεται στον context server για τη διευκόλυνση της ταυτόχρονης πολλαπλής πρόσβασης. Πέρα από την επαναχρησιμοποίηση των αισθητήρων, η χρήση ενός context server έχει το πλεονέκτημα της ανακούφισης των πελατών από εντατικές ενέργειες σε πόρους. Καθώς η πλειοψηφία των τερματικών συσκευών που χρησιμοποιούνται σε συστήματα επίγνωσης πλαισίου είναι κινητές συσκευές με περιορισμούς σε υπολογιστική δύναμη, χώρο δίσκου κ.λπ. αυτό το θέμα αποτελεί μία σημαντική πτυχή. Το μειονέκτημα είναι ότι κατά το σχεδιασμό ενός συστήματος που βασίζεται σε αρχιτεκτονική πελατών/εξυπηρετητή πρέπει να εξεταστεί με ιδιαίτερη προσοχή η επιλογή κατάλληλων πρωτόκολλων, να ληφθεί υπόψη η απόδοση δικτύου, η ποιότητα των παραμέτρων υπηρεσίας κλπ.

Όπως για την ανάκτηση πλαισίου έτσι και για τη διαχείρισή του μπορούν να ακολουθηθούν τα ακόλουθα μοντέλα:

1. Widgets: Η ονομασία προκύπτει από τα ομώνυμα GUI στοιχεία. Πρόκειται για συστατικά λογισμικού που παρέχουν διεπαφές για ένα αισθητήρα υλικού. Τα widgets κρύβουν τις πληροφορίες αίσθησης χαμηλού επιπέδου και διευκολύνουν την ανάπτυξη εφαρμογών εξαιτίας της επαναχρησιμοποίησής τους. Λόγω της ενθυλάκωσης στα widgets είναι πιθανό να γίνει ανταλλαγή widgets τα οποία

παρέχουν τον ίδιο τύπο δεδομένων πλαισίου. Τα widget ελέγχονται συνήθως από κάποιον διαχειριστή widget.

2. Networked services: Αυτή η πιο ευέλικτη προσέγγιση, μοιάζει με την αρχιτεκτονική του εξυπηρετητή πλαισίου. Αντί για έναν κεντρικοποιημένο και καθολικό διαχειριστή widget, χρησιμοποιούνται τεχνικές ανακάλυψης για την εύρεση υπηρεσιών δικτύου.
3. Blackboard model: Σε αντίθεση με την διεργασιο-στραφή προσέγγιση των widgets και το υπηρεσιο-στραφές μοντέλο, το blackboard model αναπαριστά μία δεδομένο-στραφή προσέγγιση. Σε αυτή την ασύμμετρη προσέγγιση οι διεργασίες καταγράφουν μηνύματα σε ένα διαμοιραζόμενο μέσο, το blackboard, και εγγράφονται σε αυτό ώστε να λαμβάνουν ειδοποιήσεις όποτε συμβαίνει κάποιο γεγονός. Πλεονεκτήματα αυτής της προσέγγισης είναι η απλότητα της προσθήκης νέων πηγών πληροφορίας πλαισίου και η εύκολη διαχείριση. Αντίθετα, μειονέκτημα είναι η ανάγκη ύπαρξης κεντρικοποιημένου εξυπηρετητή που θα εγκατασταθεί το blackboard και η έλλειψη αποδοτικής επικοινωνίας, αφού χρειάζονται δύο βήματα ανά επικοινωνία [2].

Όπως αναφέρθηκε και πιο πάνω, είναι απαραίτητος ο διαχωρισμός ανακάλυψης και χρήσης πληροφορίας πλαισίου ώστε να βελτιωθεί η επεκτασιμότητα και η επαναχρησιμοποίηση των συστημάτων. Η ακόλουθη αρχιτεκτονική, όπως απεικονίζεται στον Πίνακα 1, εμπλουτίζει τα στρώματα για αναγνώριση και χρήση πληροφορίας πλαισίου με λειτουργικότητα μετάφρασης και ερμηνείας της πληροφορίας.

Πίνακας 1: Επίπεδα λειτουργικότητας για συστήματα επίγνωσης πλαισίου

Εφαρμογή
Αποθήκευση/Διαχείριση
Προεπεξεργασία
Ανάκτηση ακατέργαστων δεδομένων
Αισθητήρες

Το 1ο επίπεδο αποτελείται από ένα σύνολο από διαφορετικούς αισθητήρες. Είναι αξιοσημείωτο ότι ο όρος «αισθητήρας» δεν αναφέρεται μόνο στην ανακάλυψη υλικού (hardware sensing) αλλά επίσης σε κάθε πηγή πληροφοριών που ενδέχεται να παρέχει χρήσιμες πληροφορίες πλαισίου. Όσον αφορά τον τρόπο που γίνεται η ανακάλυψη της πληροφορίας, οι αισθητήρες μπορούν να κατηγοριοποιηθούν στις ακόλουθες τρεις κατηγορίες:

1. Φυσικοί αισθητήρες: Αποτελούν την πιο συχνά χρησιμοποιούμενη κατηγορία αισθητήρων. Υπάρχει πληθώρα αισθητήρων υλικού οι οποίοι έχουν τη δυνατότητα να ανιχνευθούν σχεδόν όλα τα φυσικά δεδομένα. Ο Πίνακας 2 αποτυπώνει κάποια παραδείγματα φυσικών αισθητήρων, ενώ και στην εικόνα 1 και στην εικόνα 2 απεικονίζονται παραδείγματα φυσικών αισθητήρων.
2. Εικονικοί αισθητήρες: Οι εικονικοί αισθητήρες αντλούν τη πληροφορία πλαισίου από λογισμικό ή από υπηρεσίες. Για παράδειγμα, είναι πιθανό να καθοριστεί η θέση ενός εργαζόμενου όχι μόνο με τη χρήση συστημάτων εντοπισμού αλλά επίσης με τη βοήθεια ενός εικονικού αισθητήρα όπως ένα ηλεκτρονικό ημερολόγιο, ένα σύστημα ταξιδιωτικών κρατήσεων, το ηλεκτρονικό ταχυδρομείο

κλπ. για πληροφορίες τοποθεσίας. Άλλα χαρακτηριστικά πληροφορίας πλαισίου που υπάρχει δυνατότητα να αναγνωριστούν από εικονικούς αισθητήρες είναι η δραστηριότητα του χρήστη ελέγχοντας την κίνηση του ποντικιού ή την είσοδο του πληκτρολογίου κλπ.

3. Λογικοί αισθητήρες: Αυτοί οι αισθητήρες εκμεταλλεύονται κάποιες πηγές πληροφορίας και χρησιμοποιώντας φυσικούς και εικονικούς αισθητήρες σε συνδυασμό με επιπλέον πληροφορίες που ανακτούν από βάσεις δεδομένων και άλλες πηγές εκτελούν υψηλότερου επιπέδου διεργασίες. Για παράδειγμα ένας λογικός αισθητήρας μπορεί να κατασκευαστεί με τέτοιο τρόπο ώστε να εντοπίζει την ακριβή θέση ενός εργαζόμενου με βάση τις εισόδους που έχει κάνει σε σταθερούς υπολογιστές, χρησιμοποιώντας παράλληλα μία βάση δεδομένων ώστε να αντιστοιχίζει συσκευές με πληροφορίες τοποθεσίας.

Πίνακας 2: Συχνά χρησιμοποιούμενοι τύποι αισθητήρων

Τύπος πλαισίου	Διαθέσιμοι αισθητήρες
Φως	Αισθητήρες IR,UV
Οπτικό πλαίσιο	Κάμερες
Ήχος	Μικρόφωνα
Κίνηση, Επιτάχυνση	Αισθητήρες ανίχνευσης κίνησης
Τοποθεσία	GPS,GSM
Αφή	Αισθητήρες αφής σε κινητές συσκευές
Θερμοκρασία	Θερμόμετρα
Φυσικά χαρακτηριστικά	Βιο-αισθητήρες για τη μέτρηση της πίεσης αίματος

Το 2ο επίπεδο είναι υπεύθυνο για την ανάκτηση ακατέργαστων δεδομένων πλαισίου (raw context data). Χρησιμοποιεί το απαραίτητο λογισμικό για την αναγνώριση των φυσικών αισθητήρων καθώς και τις κατάλληλες διεπαφές για εικονικούς και λογικούς αισθητήρες. Η διαδικασία επερώτησης για δεδομένα συνήθως υλοποιείται σε επίπεδο λογισμικού (δηλαδή σε πιο αφαιρετικό επίπεδο) επιτρέποντας την επαναχρησιμοποίηση της και διατηρώντας διαφανή προς το χρήστη τη διαδικασία πρόσβασης στο υλικό. Η χρήση διεπαφών υπεύθυνων για ίδιου τύπου πληροφορία πλαισίου επιτρέπει την αντικατάσταση του συστήματος (π.χ. αντικατάσταση ενός RFID συστήματος με GPS) χωρίς να απαιτούνται πολλές αλλαγές σε αυτό και τα από πάνω επίπεδα.

Το επίπεδο προεπεξεργασίας δεν είναι υλοποιημένο σε όλα τα συστήματα επίγνωσης πλαισίου αλλά παρέχει χρήσιμες πληροφορίες αν τα ακατέργαστα δεδομένα δεν διατηρούν καμία συνοχή. Το επίπεδο αυτό είναι υπεύθυνο για την λειτουργικότητα της μετάφρασης και ερμηνείας της πληροφορίας πλαισίου. Οι αισθητήρες που λειτουργούν στα πιο κάτω επίπεδα συχνά επιστρέφουν τεχνικά δεδομένα τα οποία δεν είναι κατάλληλα για χρήση από τους σχεδιαστές της εφαρμογής. Έτσι, αυτό το επίπεδο αναδεικνύει τα αποτελέσματα του επιπέδου 2 στα υψηλότερα και πιο αφηρημένα επίπεδα. Οι μετατροπές οι οποίες κάνει περιλαμβάνουν τεχνικές εξαγωγής και ποσοτικοποίησης δεδομένων. Για παράδειγμα, η ακριβής GPS θέση ενός ανθρώπου μπορεί να μην έχει αξία για μία εφαρμογή, ενώ ο χώρος στον οποίο βρίσκεται αυτός (αίθουσα συνεδριάσεων, σινεμά κλπ.) μπορεί να έχει.

Σε συστήματα επίγνωσης πλαισίου τα οποία αποτελούνται από πολλές διαφορετικές πηγές δεδομένων, οι απλές πληροφορίες μπορούν να συνδυαστούν με υψηλού επιπέδου πληροφορίες σε αυτό το επίπεδο. Η διαδικασία αυτή ονομάζεται συσσώματωση (Aggregation) ή σύνθεση (Composition). Μία μόνο τιμή ενός αισθητήρα συχνά δεν είναι σημαντική για μία εφαρμογή, ενώ συνδυασμός πληροφοριών μπορεί να αποδειχθεί χρησιμότερος και πιο ακριβής. Υπό αυτό το πρίσμα ένα σύστημα είναι σε θέση να καθορίσει αν για παράδειγμα ένας πελάτης βρίσκεται σε κλειστό ή σε ανοιχτό χώρο, αναλύοντας διάφορα φυσικά δεδομένα, όπως η θερμοκρασία και το φως ή αν παρακολουθεί ένα συνέδριο αναλύοντας το επίπεδο θορύβου και τη τοποθεσία του. Για να λειτουργήσει σωστά η ανάλυση αυτή, επιστρατεύεται η χρήση πολλών στατιστικών μεθόδων, ενώ συχνά απαιτείται και κάποια φάση εκπαίδευσης (training phase).

Αν και αυτή η αφηρημένη λειτουργικότητα θα μπορούσε, να υλοποιηθεί απευθείας στο επίπεδο της εφαρμογής είναι προτιμότερο να γίνει ενθυλάκωση και να μετακινηθεί στο επίπεδο του εξυπηρετητή. Η ενθυλάκωση προάγει την επαναχρησιμοποίηση και έτσι, διευκολύνει την ανάπτυξη εφαρμογών. Επιπλέον, καθιστώντας τέτοιους συλλέκτες προσπελάσιμους από απόσταση αυξάνεται η απόδοση του δικτύου (αφού οι πελάτες πρέπει να στείλουν μόνο μία αίτηση για να έχουν πρόσβαση σε υψηλού επιπέδου δεδομένα, αντί να πρέπει να συνδεθούν σε πολλούς και διαφορετικούς αισθητήρες) και εξοικονομούνται οι, ούτως ή άλλως, περιορισμένοι πόροι των πελατών.

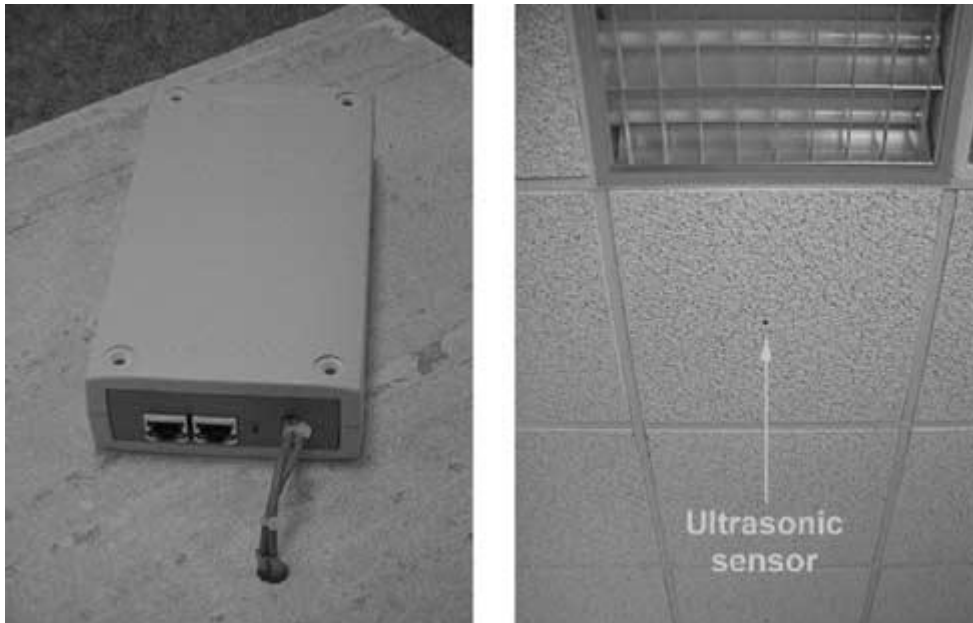
Επιπλέον, πρέπει να επιλυθούν σε αυτό το επίπεδο τυχόν συγκρούσεις δεδομένων από διαφορετικούς αισθητήρες. Για παράδειγμα, όταν ένα σύστημα ειδοποιείται αναφορικά με τη θέση ενός ατόμου, από τις συντεταγμένες του κινητού του τηλεφώνου και από μία κάμερα, ενδεχομένως να είναι δύσκολο να αποφασίσει ποιά πληροφορία πρέπει να θεωρήσει επακριβή και να χρησιμοποιήσει αυτή. Για να επιλυθούν τέτοιου είδους συγκρούσεις συχνά χρησιμοποιούνται επιπλέον δεδομένα, όπως χρονοσφραγίδες και κανόνες επίλυσης.

Το τέταρτο επίπεδο, αποθήκευση και διαχείριση, είναι υπεύθυνο για την οργάνωση των δεδομένων που έχουν συλλεχθεί και για τη προώθηση τους μέσω κάποιας διεπαφής στον πελάτη. Οι πελάτες μπορούν να έχουν πρόσβαση στα δεδομένα είτε με ασύγχρονο, ή με σύγχρονο τρόπο. Κατά τη σύγχρονη επικοινωνία ο πελάτης επερωτά τον εξυπηρετητή για αλλαγές στα δεδομένα μέσω κλήσης συγκεκριμένων μεθόδων (rolling technique). Δηλαδή, στέλνει ένα μήνυμα ζητώντας κάποια δεδομένα και σταματά έως ότου λάβει απάντηση. Ο ασύγχρονος τρόπος δουλεύει με εγγραφή σε γεγονότα (subscriptions). Κάθε πελάτης εγγράφεται στα γεγονότα που τον ενδιαφέρουν και όταν συμβεί κάποιο από αυτά ο πελάτης ενημερώνεται με ένα μήνυμα ή εμπλέκεται τότε κλήση κάποιας μεθόδου. Στη πλειοψηφία των περιπτώσεων η ασύγχρονη προσέγγιση είναι καταλληλότερη, εξαιτίας των ραγδαίων αλλαγών στο πλαίσιο. Η τεχνική rolling απαιτεί περισσότερους πόρους καθώς ο πελάτης ζητάει αρκετά συχνά δεδομένα πλαισίου και η εφαρμογή πρέπει να ελέγξει από μόνη της αν υπάρχει αλλαγή στο πλαίσιο, χρησιμοποιώντας κάποιο είδος ιστορικό προηγούμενων καταστάσεων.

Ο πελάτης αποτελεί ουσιαστικά το πέμπτο επίπεδο, το επίπεδο εφαρμογής. Εδώ υλοποιείται η πραγματική αντίδραση σε διάφορα γεγονότα. Κάποιες φορές η ανακάλυψη πληροφοριών, η διαχείριση πληροφορίας πλαισίου και η σημασιολογία αυτής θεωρείται ως ένας πράκτορας, ο οποίος επικοινωνεί με τον εξυπηρετητή πλαισίου και συμπεριφέρεται ως ένα επιπλέον επίπεδο ανάμεσα στην προεπεξεργασία και το επίπεδο εφαρμογής. Ένα παράδειγμα στα πλαίσια της λογικής πλαισίου στον πελάτη είναι η οθόνη σε κινητές συσκευές, όπου όταν ο αισθητήρας φωτός αντιληφθεί κακό φωτισμό, μπορεί να εμφανίζει κείμενο με υψηλότερη αντίθεση χρωμάτων [2].



Εικόνα 1: Μία μονάδα για εντοπισμό θέσης τύπου Bat



Εικόνα 2: Εγκατάσταση ενός αισθητήρα

2.3.2.2 Μοντέλα Πληροφορίας Πλαισίου (Context Model)

Ένα μοντέλο πληροφορίας πλαισίου είναι απαραίτητο ώστε να καθοριστούν και να αποθηκεύονται δεδομένα που θεωρούνται πληροφορία πλαισίου σε επεξεργάσιμη μορφή. Η ανάπτυξη χρήσιμων οντολογιών που να καλύπτουν ένα μεγάλο πεδίο πιθανών πλαισίων είναι αρκετά δύσκολη. Ακολούθως συνοψίζονται οι πιο σχετικές προσεγγίσεις για μοντελοποίηση πλαισίου, ως προς τις δομές δεδομένων που

χρησιμοποιήθηκαν για την αναπαράσταση και την ανταλλαγή πληροφορίας πλαισίου στο σύστημα.

1. Μοντέλα ζευγαριών κλειδιού τιμής (Key-Value models): Αυτά τα μοντέλα αναπαριστούν την απλούστερη δομή δεδομένων για μοντελοποίηση πλαισίου. Χρησιμοποιούνται συχνά σε διάφορα πρότυπα υπηρεσιών, όπου τα ζευγάρια κλειδί-τιμή χρησιμοποιούνται για τη περιγραφή των δυνατοτήτων μίας υπηρεσίας. Έπειτα, η ανακάλυψη υπηρεσίας επιτυγχάνεται εφαρμόζοντας αλγορίθμους αντιστοίχισης (matching algorithms) που χρησιμοποιούν αυτά τα ζευγάρια.
2. Μοντέλα μαρκαρίσματος (Markup scheme models): Όλα τα μοντέλα μαρκαρίσματος χρησιμοποιούν μία ιεραρχική δομή δεδομένων η οποία αποτελείται από ταμπέλες μαρκαρίσματος με χαρακτηριστικά και περιεχόμενο. Τα προφίλ αναπαριστούν τυπικά σχήματα μοντέλων μαρκαρίσματος. Τυπικά παραδείγματα για τέτοια προφίλ είναι τα CC/PP και UAProf τα οποία κωδικοποιούνται σε RDF/S.
3. Γραφικά μοντέλα (Graphical Models): Υπάρχουν διάφορες προσεγγίσεις που στηρίζονται στη γλώσσα UML, η οποία ταιριάζει στη μοντελοποίηση πλαισίου.
4. Αντικειμενοστραφή μοντέλα: Η μοντελοποίηση πλαισίου με τη χρήση αντικειμενοστραφών τεχνικών δίνει τη δυνατότητα χρήσης όλων των δυνατοτήτων της αντικειμενοστράφειας (ενθυλάκωση, κληρονομικότητα, επαναχρησιμοποίηση). Οι υπάρχουσες προσεγγίσεις χρησιμοποιούν διάφορα αντικείμενα για την αναπαράσταση διαφορετικών τύπων πλαισίου (όπως θερμοκρασία, τοποθεσία κλπ.) στα οποία ενθυλακώνουν τις πληροφορίες αναπαράστασης και επεξεργασίας της πληροφορίας πλαισίου. Η πρόσβαση στη πληροφορία πλαισίου επιτυγχάνεται από καλά ορισμένες διεπαφές (interfaces).
5. Λογικά μοντέλα (Logic-based models): Τα λογικά μοντέλα έχουν υψηλό βαθμό φορμαλισμού. Τυπικά, χρησιμοποιούνται γεγονότα, εκφράσεις κανόνες για τον προσδιορισμό ενός μοντέλου. Έπειτα, χρησιμοποιείται ένα σύστημα που βασίζεται σε κανόνες λογικής για να διαχειριστεί τους προαναφερθέντες όρους, το οποίο επιτρέπει τη προσθήκη, την ενημέρωση ή τη διαγραφή γεγονότων. Επιπλέον, μπορεί να χρησιμοποιηθεί μία διαδικασία συμπερασμού για να παράγει νέα γεγονότα με βάση ήδη υπάρχοντες κανόνες του συστήματος. Η πληροφορία πλαισίου πρέπει να αναπαρίσταται με τυπική μορφή, δηλαδή σαν δεδομένα.
6. Μοντέλα οντολογιών (Ontology-based models) : Οι οντολογίες αναπαριστούν τη περιγραφή εννοιών και σχέσεων. Άρα, οι οντολογίες αποτελούν ένα πολλά υποσχόμενο εργαλείο για τη μοντελοποίηση πληροφορίας πλαισίου, λόγω της υψηλής και τυπικής εκφραστικότητάς τους καθώς επίσης και εξαιτίας των δυνατοτήτων εφαρμογής τεχνικών συμπερασμού. Διάφορα context-aware frameworks, όπως η αρχιτεκτονική CoBRa, χρησιμοποιούν οντολογίες ως το βασικό εργαλείο μοντελοποίησης πληροφορίας πλαισίου.

Από τα παραπάνω μοντέλα τα ontology-based models είναι τα πιο εκφραστικά και πληρούν τις περισσότερες απαιτήσεις. Βασικές απαιτήσεις και στόχοι που πρέπει να θέτει μία οντολογία πλαισίου είναι οι εξής:

1. Απλότητα: οι χρησιμοποιούμενες εκφράσεις και σχέσεις πρέπει να είναι όσο πιο απλές γίνεται, ώστε να απλοποιείται η δουλειά των προγραμματιστών.
2. Ευελιξία και επεκτασιμότητα: Η οντολογία πρέπει να υποστηρίζει την κατά τον δυνατόν απλούστερη προσθήκη νέων στοιχείων πλαισίου και σχέσεων

3. Γενικότητα: Η οντολογία δεν πρέπει να περιορίζεται σε ειδικού τύπου μονάδες πληροφορίας πλαισίου (context atoms) αλλά να υποστηρίζει διαφορετικά είδη πλαισίου.
4. Εκφραστικότητα: Η οντολογία πρέπει να επιτρέπει όσο το δυνατόν περισσότερες καταστάσεις πλαισίου

Πολλά εργαλεία είναι διαθέσιμα για τον ορισμό δηλωτικών αναπαραστάσεων, την έκδοση, και τον διαμοιρασμό οντολογιών που αναπτύσσονται από το W3C. Τέτοια εργαλεία είναι η RDF και η OWL. Μία μονάδα πληροφορίας πλαισίου μπορεί να περιγραφεί με ένα σύνολο από χαρακτηριστικά. Τα πιο συνηθισμένα είναι:

1. Τύπος πλαισίου (context type): Ο τύπος πλαισίου αναφέρεται στη κατηγορία πλαισίου όπως θερμοκρασία, χρόνος, ταχύτητα κλπ. Τέτοιου είδους πληροφορία μπορεί να χρησιμοποιηθεί ως παράμετρος για κάποιου είδους επερώτηση, ή για εγγραφή σε υπηρεσίες. Για παράδειγμα μία μέθοδος για εγγραφή σε ενημερώσεις της θερμοκρασίας είναι η ακόλουθη: subscribe On Update ('temperature'). Είναι σημαντικό να χρησιμοποιούνται τύποι ονομάτων με φυσική σημασία, καθώς, όπως το σύστημα επεκτείνεται κάποια ονόματα μπορεί να μην είναι μοναδικά πλέον. Για παράδειγμα, η τοποθεσία ως τύπος πλαισίου μπορεί να ανήκει σε μία κινητή συσκευή ή ένα χρήστη. Μία λύση για δημιουργία καλά δομημένων τύπων ονομάτων είναι η χρήση κλιμακούμενων ονομάτων, όπως φαίνεται και στο Πίνακας 3.
2. Η τιμή της πληροφορίας πλαισίου (context value): Context value σημαίνει τα ακατέργαστα δεδομένα που συγκεντρώνονται από τον αισθητήρα. Η μονάδα μέτρησης εξαρτάται από το τύπο του πλαισίου και τον αισθητήρα. Π.χ. βαθμοί Κελσίου, χιλιόμετρα ανά ώρα κλπ.

Πίνακας 3: Χρήση κλιμακούμενων ονομάτων για μεθόδους

<i>Context type</i>	<i>Context</i>
Environment:Temperature	Cold
Environment:Temperature	Normal
Environment:Temperature	Hot
Environment:Light:Source	50 Hz
Environment:Light:Source	60 Hz
Environment:Light:Source	NotAvailable
Device:Activity:Placement	AtHand
Device:Activity:Placement	NotAtHand

Στις περισσότερες περιπτώσεις το context type και value δεν είναι αρκετή πληροφορία για τη κατασκευή ενός context-aware συστήματος [2]. Επιπλέον χαρακτηριστικά που μπορεί να φανούν χρήσιμα είναι τα εξής:

1. Χρονική σφραγίδα : Αυτό το χαρακτηριστικό περιέχει μία ημερομηνία και ώρα η οποία περιγράφει τότε ο αισθητήρας ανπλήφθηκε τη πληροφορία πλαισίου. Αυτή

η χρονική σφραγίδα χρειάζεται για τη δημιουργία ιστορικού ανάκτησης πληροφορίας πλαισίου και για την επίλυση συγκρούσεων που προκύπτουν από διαφορετική αντίληψη του ίδιου πλαισίου που ενδέχεται να έχουν δύο ή περισσότεροι αισθητήρες.

2. Πηγή: Είναι ένα πεδίο που περιέχει πληροφορίες για το πώς ανακτήθηκε η πληροφορία πλαισίου. Στη περίπτωση των αισθητήρων υλικού μπορεί να περιέχει το αναγνωριστικό του αισθητήρα και να επιτρέπει σε μία εφαρμογή να προτιμά δεδομένα από αυτόν τον αισθητήρα.
3. Βαθμός εμπιστοσύνης : Το χαρακτηριστικό αυτό περιγράφει την αβεβαιότητα ως προς το τύπο του πλαισίου, καθώς δεν δίνουν όλες οι πηγές δεδομένων επακριβή πληροφορία. Για παράδειγμα δεδομένα τοποθεσίας του χρήστη εμφανίζουν ανακρίβειες ανάλογα με το εργαλείο εντοπισμού που χρησιμοποιήθηκε.

3. Swarm Intelligence: Υποδομές, Προοπτικές και Εφαρμογές

Σε αυτό το κεφάλαιο παρουσιάζονται κάποιες από τις θεωρητικές βάσεις του Swarm Intelligence. Ιδιαίτερη έμφαση δίνεται στην σχεδίαση και την υλοποίηση του αλγορίθμου PSO για διαφόρων ειδών προβλήματα βελτιστοποίησης, εφαρμογές πραγματικού κόσμου και εξόρυξης δεδομένων. Τα αποτελέσματα αναλύονται, συζητούνται και διευκρινίζονται οι δυνατότητές τους.

3.1 Εισαγωγή

Η Swarm Intelligence (SI) είναι ένα καινοτόμο, κατανεμημένο και ευφυές παράδειγμα για επίλυση προβλημάτων βελτιστοποίησης. Αρχική πηγή έμπνευσης για τον ερευνητικό τομέα της swarm intelligence ήταν παραδείγματα βιολογίας όπως τα φαινόμενα σμήνους εντόμων, σμήνους πτηνών και αγέλης σπονδυλωτών. Από το 1990 έχουν προταθεί αρκετοί αλγόριθμοι, οι οποίοι στηρίζονται σε αντιδράσεις ενός συνόλου (όπως το πέταγμα ενός σμήνους πουλιών). Τα πεδία εφαρμογής τέτοιων αλγορίθμων σχετίζονται με γνωστά και αρκετά μελετημένα προβλήματα, όπως NP-hard προβλήματα (το πρόβλημα του πλανόδιου πωλητή, προβλήματα γράφων κ.α.), προβλήματα δρομολόγησης δικτύου, συσταδοποίησης, εξόρυξης δεδομένων, χρονικού προγραμματισμού εργασιών κλπ. [20].

Ένας από τους πιο διάσημους αλγορίθμους στην ερευνητική περιοχή του swarm intelligence είναι ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (particle swarm optimization algorithm, PSO algorithm), ο οποίος ενσωματώνει συμπεριφορές συνόλου, οι οποίες έχουν παρατηρηθεί σε σμήνη πτηνών, σμήνη εντόμων, ή αγέλες ψαριών, ή ακόμα και σε κοινωνικές συμπεριφορές ανθρώπων, απ' όπου προέκυψε και η ιδέα. Ο αλγόριθμος PSO είναι εργαλείο βελτιστοποίησης στηριζόμενο σε πληθυσμό, το οποίο θα μπορούσε να μοντελοποιηθεί, να υλοποιηθεί και να εφαρμοστεί εύκολα για να επιλύσει διάφορα προβλήματα βελτιστοποίησης συναρτήσεων. Το δυνατότερο σημείο του αλγορίθμου είναι η γρήγορη σύγκλισή του ή οποία είναι καλύτερη σε σχέση με πολλούς αλγορίθμους καθολικής βελτιστοποίησης (global optimization algorithms) όπως οι γενετικοί αλγόριθμοι (Genetic Algorithms), ή ο αλγόριθμος Simulated Annealing. Για να εφαρμοστεί ο αλγόριθμος PSO επιτυχώς, ένα από τα ζητήματα κλειδιά είναι να βρούμε πως θα απεικονίσουμε τη λύση του προβλήματος σε ένα σωματίδιο, καθώς επηρεάζει άμεσα την απόδοση αλλά και κατ' επέκταση το αν είναι τελικά εφαρμόσιμος ο αλγόριθμος στο πρόβλημα [4],[20].

3.2 Κανονική Μορφή Αλγορίθμου PSO

3.2.1 Κανονικό μοντέλο

Το κανονικό μοντέλο PSO αποτελείται από ένα σμήνος σωματιδίων, τα οποία αρχικοποιούνται με ένα πλήθος τυχαίων υποψήφιων λύσεων. Αυτά κινούνται επαναληπτικά μέσα στο χώρο (d διαστάσεων) του προβλήματος, αναζητώντας νέες λύσεις, όπου η συνάρτηση f θα πληροί καθορισμένο έλεγχο καταλληλότητας (fitness function). Κάθε κόμβος έχει μια θέση η οποία αναπαρίσταται από το διάνυσμα θέσης x_i (όπου i είναι η ένδειξη του κόμβου) και μία ταχύτητα η οποία αναπαρίσταται από το διάνυσμα ταχύτητας v_i . Κάθε κόμβος θυμάται την δική του καλύτερη θέση μέχρι στιγμής σε ένα διάνυσμα $x_i^{\#}$, όπου η τιμή της j -οστής διάστασης είναι η $x_{ij}^{\#}$. Το καλύτερο διάνυσμα θέσης ανάμεσα στο σμήνος μέχρι στιγμής καταχωρείται στο διάνυσμα x^* ,

οπότε αντίστοιχα η μεταβλητή x_j^* αναπαριστά τη τιμή της j-οστής διάστασης. Κατά τον επαναληπτικό χρόνο t του αλγορίθμου, η ενημέρωση της ταχύτητας από τη παλιά τιμή στη νέα καθορίζεται από την Εξίσωση 1. Έπειτα καθορίζεται η νέα τιμή του διανύσματος θέσης ως το άθροισμα της προηγούμενης θέσης και της νέας ταχύτητας, όπως φαίνεται στην Εξίσωση 2.

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1 * (x_{ij}^{\#}(t) - x_{ij}(t)) + c_2 * r_2 * (x_j^*(t) - x_{ij}(t))$$

Εξίσωση 1: Ενημέρωση ταχύτητας κόμβου

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

Εξίσωση 2: Υπολογισμός διανύσματος θέσης

όπου w είναι ο παράγοντας αδράνειας βάρους (inertia weight), r_1 και r_2 είναι δύο τυχαίοι αριθμοί, οι οποίοι χρησιμοποιούνται για να διατηρούν τη ποικιλία του πληθυσμού και είναι ομοιόμορφα κατανεμημένοι στο διάστημα $[0,1]$ για την j-οστή διάσταση του i-οστού κόμβου [4]. Η c_1 είναι μία θετική σταθερά γνωστή ως συντελεστής προσωπικής συνιστώσας, ενώ η c_2 είναι ο συντελεστής κοινωνικής συνιστώσας. Από την Εξίσωση 1, κάθε κόμβος επιλέγει που θα κινηθεί στη συνέχεια, αναλογιζόμενος τη δική του πρότερη εμπειρία, δηλαδή τη καλύτερη έως τώρα θέση, αλλά και την πείρα του πιο πετυχημένου κόμβου στο σμήνος. Στο μοντέλο σμήνους σωματιδίων το οποίο μελετάμε, το κάθε σωματίδιο αναζητά για λύσεις στο χώρο του προβλήματος με εύρος $[-s,s]$. Για να καθοδηγήσουμε τα σωματίδια αποτελεσματικά στο χώρο αναζήτησης, η μέγιστη απόσταση μετακίνησης για μία επανάληψη πρέπει να περιοριστεί στο διάστημα μέγιστης ταχύτητας $[-v_{max}, v_{max}]$ όπως φαίνεται στην Εξίσωση 3:

$$v_{ij} = \text{sign}(v_{ij}) * \min(|v_{ij}|, v_{max})$$

Εξίσωση 3: Περιορισμός μέγιστης ταχύτητας ανά χρονική στιγμή

Η τιμή του v_{max} είναι $p*s$, με $0.1 \leq p \leq 1.0$ και συνήθως επιλέγεται να είναι s . Ο Πίνακας 4 παρουσιάζει σε ψευδοκώδικα τον αλγόριθμο.

Πίνακας 4: Ψευδοκώδικας για το κανονικό μοντέλο του αλγορίθμου PSO

1	Αρχικοποίηση μεγέθους του σμήνους s και άλλων παραμέτρων
2	Αρχικοποίηση των θέσεων και των ταχυτήτων των σωματιδίων με τυχαίο τρόπο
3	Όσο δεν ισχύει το κριτήριο τερματισμού κάνε
4	$t=t+1$;
5	Υπολόγισε το fitness για κάθε σωματίδιο
6	$\mathbf{x}^* = \operatorname{argmin}_{i=1}^n (f(\mathbf{x}^*(t-1)), f(\mathbf{x}_1(t)), f(\mathbf{x}_2(t)), \dots, f(\mathbf{x}_i(t)), \dots, f(\mathbf{x}_n(t)))$;
7	Για $i=1$ έως n
8	$\mathbf{x}_i^\#(t) = \operatorname{argmin}_{i=1}^n (f(\mathbf{x}_i^\#(t-1)), f(\mathbf{x}_i(t)))$;
9	Για $j = 1$ έως Dimension
10	Ενημέρωσε την j th dimension τιμή των \mathbf{x}_i και \mathbf{v}_i σύμφωνα με την Εξίσωση 1, την Εξίσωση 2 και την Εξίσωση 3
11	Επόμενο j
12	Επόμενο i
13	Τέλος του όσο

Τα κριτήρια τερματισμού του αλγορίθμου είναι τα ακόλουθα [4]:

1. Μέγιστο πλήθος επαναλήψεων: η διαδικασία βελτιστοποίησης τερματίζει έπειτα από ένα καθορισμένο πλήθος επαναλήψεων, για παράδειγμα, 1000 επαναλήψεις.
2. Πλήθος επαναλήψεων χωρίς βελτίωση: η διαδικασία βελτιστοποίησης τερματίζει έπειτα από ένα καθορισμένο πλήθος επαναλήψεων χωρίς βελτίωση.
3. Ελάχιστη απόκλιση αντικειμενικής συνάρτησης: το λάθος ανάμεσα στην τιμή που βρήκαμε από την αντικειμενική συνάρτηση και τη καλύτερη τιμή της συνάρτησης f είναι μικρότερο από ένα προκαθορισμένο προσδοκώμενο κατώφλι.

3.2.2 Οι παράμετροι του particle swarm optimization αλγορίθμου

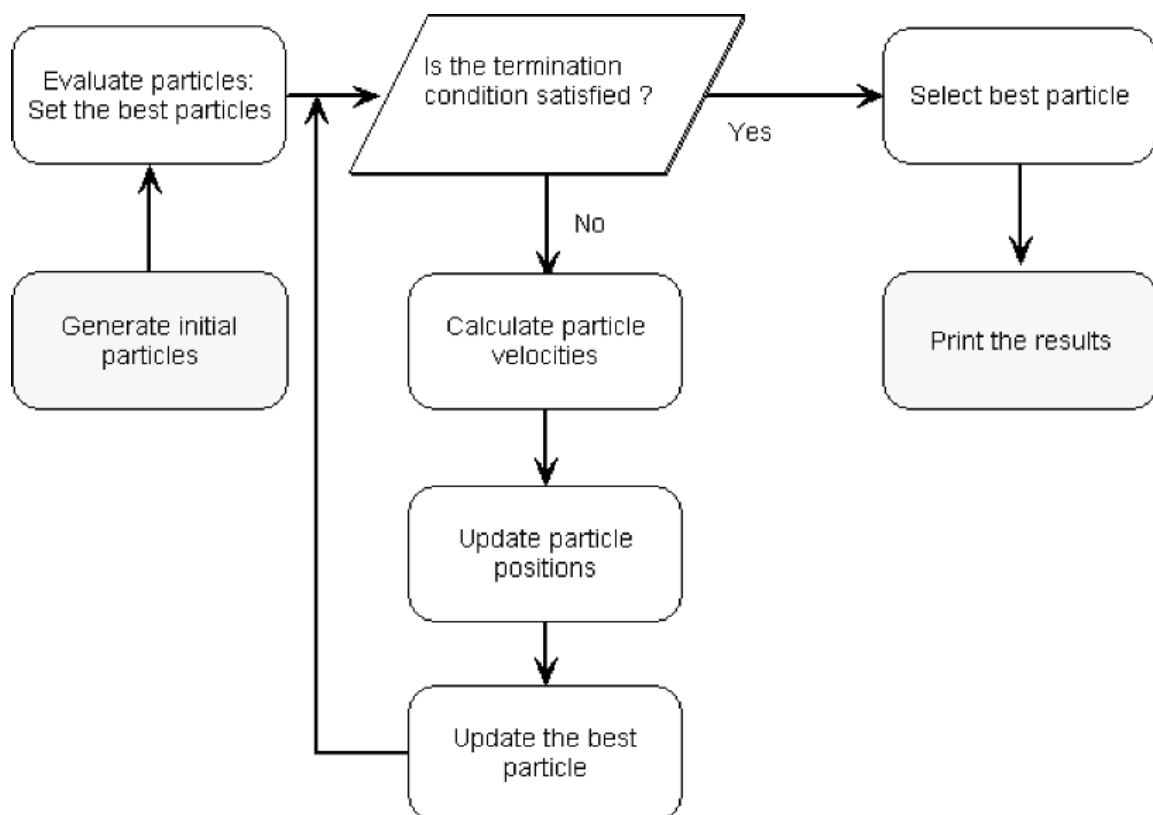
Ο ρόλος της αδράνειας βάρους (inertia weight) w στην Εξίσωση 1, θεωρείται κρίσιμος για τη συμπεριφορά σύγκλισης του αλγορίθμου PSO. Η αδράνεια βάρους έχει σαν ρόλο να ελέγχει την επιρροή που ασκούν προηγούμενες χρονικά τιμές ταχύτητας στην τρέχουσα. Άρα, η παράμετρος w ρυθμίζει τη σχέση ανάμεσα στις ικανότητες ευρείας και τοπικής αναζήτησης του σμήνους. Εάν η αδράνεια βάρους έχει μεγάλη τιμή, διευκολύνει την ευρεία αναζήτηση (άρα την αναζήτηση σε νέες περιοχές), ενώ μία μικρή τιμή έχει τη τάση να διευκολύνει τοπικές αναζητήσεις. Η κατάλληλη τιμή για την αδράνεια βάρους w συνήθως προσφέρει ισορροπία ανάμεσα στις δύο αυτές ικανότητες και κατά συνέπεια, έχει ως αποτέλεσμα τη μείωση του πλήθους των επαναλήψεων που χρειάζονται για να εντοπίσουμε τη βέλτιστη λύση. Αρχικά, η αδράνεια βάρους καθορίζεται σαν σταθερά. Όμως κάποια πειραματικά αποτελέσματα δείχνουν ότι είναι καλύτερο αρχικά να έχει αυτή η παράμετρος μεγάλη τιμή και σταδιακά να τη μειώνει έτσι ώστε να παίρνει βελτιωμένες λύσεις. Έτσι, μια αρχική τιμή περίπου 1,2 και σταδιακά μειούμενη στο 0

μπορεί να θεωρηθεί ως καλή επιλογή για τη παράμετρο w . Μία καλύτερη μέθοδος είναι να χρησιμοποιήσουμε κάποιες προσαρμοσμένες προσεγγίσεις (π.χ. ελεγκτής ασάφειας, Fuzzy controller), στις οποίες οι παράμετροι μπορούν να ρυθμιστούν ανάλογα με το πρόβλημα.

Οι παράμετροι c_1 και c_2 στην Εξίσωση 1 δεν είναι κρίσιμες για τη σύγκλιση του αλγορίθμου PSO. Όμως, κατάλληλη ρύθμιση των παραμέτρων αυτών μπορεί να έχει σαν αποτέλεσμα γρηγορότερη σύγκλιση και αποφυγή εγκλωβισμού σε τοπικό ελάχιστο. Οι τιμές που χρησιμοποιούνται συνήθως είναι $c_1=c_2=2$ αλλά κάποια πειραματικά αποτελέσματα δείχνουν ότι για $c_1=c_2=1.49$ θα μπορούσαμε να πάρουμε ακόμα καλύτερα αποτελέσματα [4]. Τέλος, ίσως θα ήταν καλύτερο, σύμφωνα με άλλες μελέτες αν η γνωσιακή παράμετρος (cognitive parameter) c_1 είχε μεγαλύτερη τιμή από τη κοινωνική παράμετρο (social parameter) c_2 με τον περιορισμό $c_1+c_2 \leq 4$.

3.2.3 Τοπολογίες

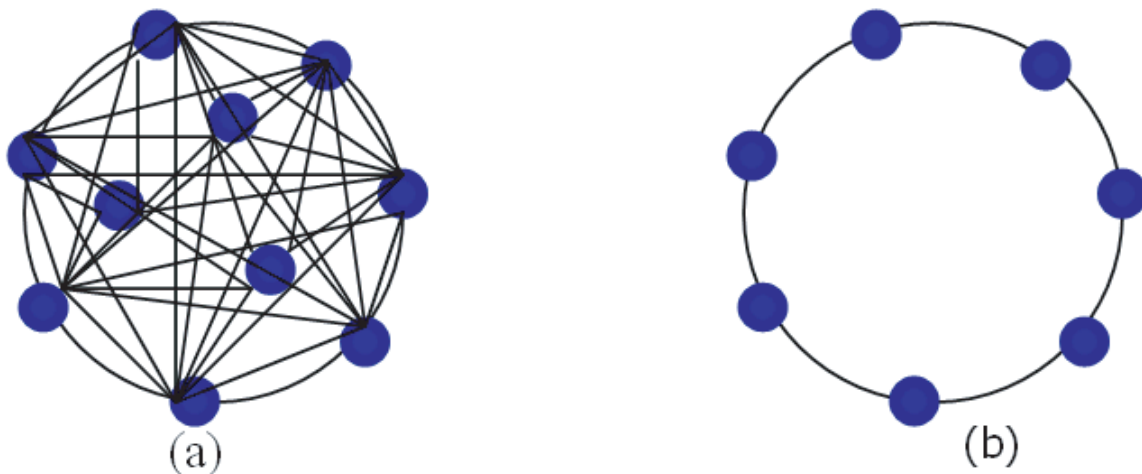
Το βασικό σχήμα του αλγορίθμου PSO αναπαρίσταται στην εικόνα 3. Ο αλγόριθμος PSO μπορεί να μελετηθεί ως ένα σύνολο από διανύσματα των οποίων οι τροχιές ταλαντεύονται γύρω από μία περιοχή καθορισμένη από την προηγούμενη καλύτερη θέση τόσο του συγκεκριμένου σωματιδίου, όσο και κάποιων γειτόνων του. Υπάρχουν διάφορες τοπολογίες γειτνίασης που χρησιμοποιούνται για να αναγνωρίσουν ποιιά σωματίδια μπορούν να επηρεάσουν ένα από αυτά. Οι πιο συνηθισμένες είναι γνωστές ως $gbest$ και $lbest$ [5].



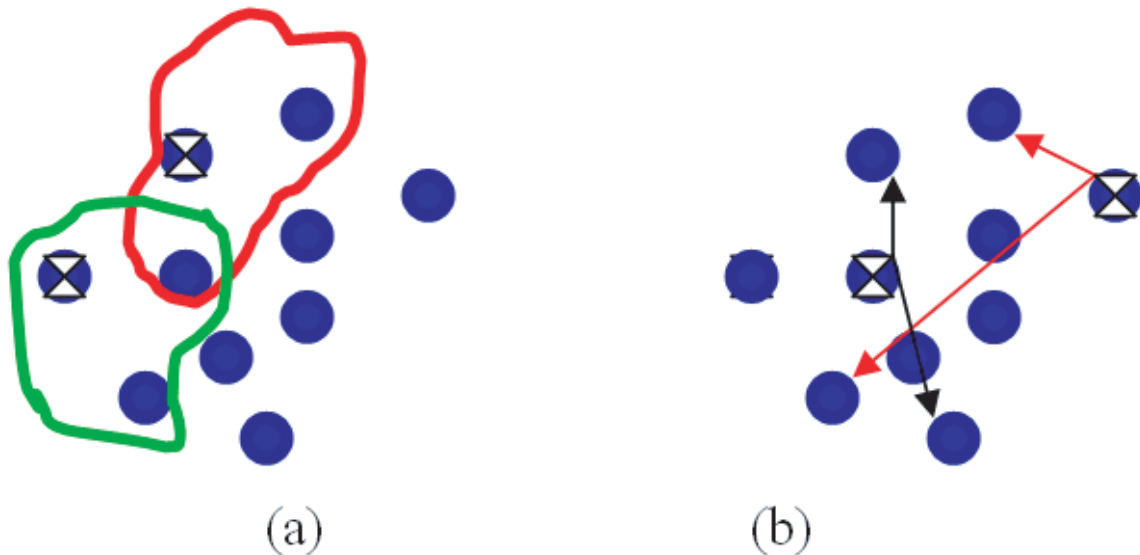
Εικόνα 3: Η βασική δομή του PSO

Στο σμήνος προτίμησης καθολικά καλύτερου σωματιδίου (*gbest swarm*), η τροχιά κάθε ενός σωματιδίου επηρεάζεται από τη καλύτερη τιμή σε ολόκληρο το σμήνος. Θεωρείται δεδομένο ότι σμήνη *gbest* συγκλίνουν γρήγορα καθώς όλα τα σωματίδια έλκονται ταυτόχρονα στο καλύτερο κομμάτι του χώρου αναζήτησης. Όμως, αν το καθολικό βέλτιστο δεν είναι κοντά στο καλύτερο σωματίδιο, μπορεί να είναι αδύνατο για το σμήνος να ερευνήσει άλλες περιοχές και συνεπώς, το σμήνος μπορεί να εγκλωβιστεί σε ένα τοπικό μέγιστο [5].

Στο σμήνος προτίμησης τοπικά καλύτερου σωματιδίου (*lbest swarm*), κάθε μονάδα επηρεάζεται από ένα μικρότερο αριθμό των γειτόνων του (τα οποία είναι τα γειτονικά μέλη στο πίνακα σμήνους). Τυπικά, οι γειτονίες *lbest* περιλαμβάνουν δύο γείτονες: έναν στα δεξιά και έναν στα αριστερά. Αυτού του τύπου τα σμήνη συγκλίνουν πιο αργά, αλλά έχουν τη μεγαλύτερη πιθανότητα να εντοπίσουν το καθολικό βέλτιστο. Το *lbest* σμήνος έχει τη δυνατότητα να κυλά γύρω από τοπικά βέλτιστα και υποσμήνη του έχουν τη δυνατότητα να ερευνούν διαφορετικά βέλτιστα. Μία γραφική αναπαράσταση ενός *gbest* και ενός *lbest* σμήνους αντίστοιχα απεικονίζεται στην εικόνα 4. Αν θεωρήσουμε κοινωνικές και γεωγραφικές γειτονίες όπως παρουσιάζονται στην εικόνα5, τότε και το *gbest* και το *lbest* μπορούν να θεωρηθούν ως μορφές κοινωνικής γειτονιάς [5].

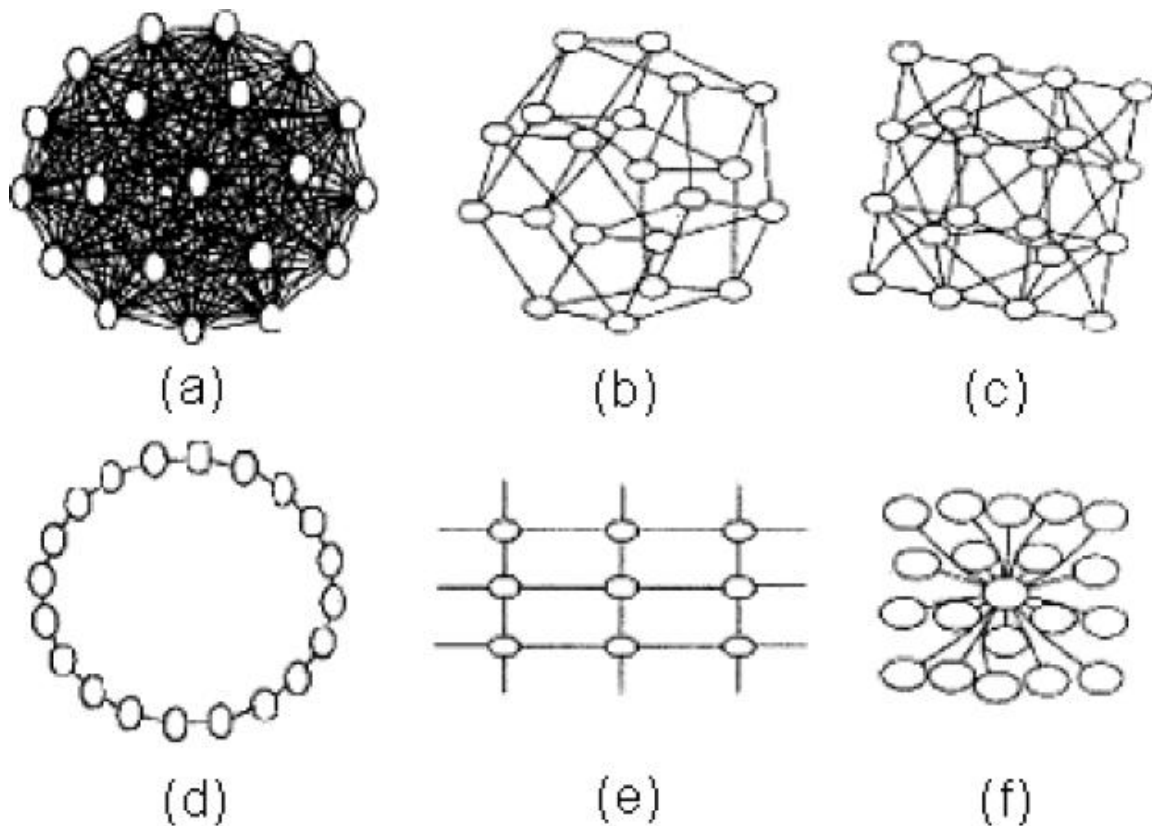


Εικόνα 4: Γραφική αναπαράσταση ενός (α)*gbest swarm* (β) *lbest swarm*



Εικόνα 5: Παραδείγματα (α) γεωγραφικής γειτονιάς (β) κοινωνικής γειτονιάς

Οι Kennedy και Mendes μελέτησαν διάφορες τοπολογίες πληθυσμού ως προς την απόδοση του αλγορίθμου PSO. Διάφορες ιδέες γειτνίασης θα μπορούσαν να είναι επιθυμητές. Θα μπορούσε να τηρηθεί μια χωρική γειτνίαση, όπου η Ευκλείδεια απόσταση ανάμεσα σε δύο σωματίδια καθορίζεται από τις θέσεις αυτών των σωματιδίων, ή θα μπορούσε να τηρηθεί μία κοινωνική μετρική ως γειτονιά. Οι διαφορετικές προσεγγίσεις για γειτονίες οδηγούν σε διαφορετικές τοπολογίες γειτνίασης. Διαφορετικές τοπολογίες γειτνίασης επηρεάζουν κυρίως την ικανότητα επικοινωνίας ανάμεσα στους κόμβους και επομένως, την απόδοση του συνόλου. Διάφορες τέτοιες τοπολογίες απεικονίζονται στην εικόνα 6. Στη περίπτωση μίας καθολικής γειτονιάς, η δομή είναι ένα πλήρως συνδεδεμένο δίκτυο όπου κάθε κόμβος έχει πρόσβαση στις καλύτερες θέσεις των άλλων κόμβων (Εικόνα 6 a). Στις τοπικές γειτονίες υπάρχουν περισσότερες επιλογές. Στη τοπολογία von Neumann (Εικόνα 6 b) γείτονες από πάνω, κάτω, δεξιά και αριστερά συνδέονται με τον κόμβο σε ένα δισδιάστατο πλέγμα. Η εικόνα 6 e απεικονίζει ένα μέρος της τοπολογίας von Neumann στο επίπεδο (σε 2 διαστάσεις). Στη τοπολογία πυραμίδας, τριγωνικά πλαίσια τριών διαστάσεων τυποποιούνται όπως φαίνεται στην εικόνα 6 c. Στην εικόνα 6 d απεικονίζεται μία κοινή δομή για τοπική γειτονιά, η τοπολογία κύκλου, όπου οι κόμβοι απέχουν πολύ μεταξύ τους (σύμφωνα με τη δομή των γράφων και όχι σύμφωνα με την Ευκλείδεια απόσταση μεταξύ τους) και είναι ανεξάρτητοι μεταξύ τους, αλλά οι γείτονες είναι στενά συνδεδεμένοι. Τέλος, στην εικόνα 6 f φαίνεται μία δομή γνωστή ως τοπολογία τροχού (ή αστέρα), η οποία έχει μια πιο ιεραρχική δομή, επειδή όλα τα μέλη της γειτονιάς συνδέονται σε έναν κόμβο «ηγέτη». Έτσι, όλη η πληροφορία πρέπει να περάσει από το κόμβο «ηγέτη», ο οποίος μετά συγκρίνει τις αποδόσεις των άλλων σωματιδίων [4].



Εικόνα 6: Παραδείγματα τοπολογιών

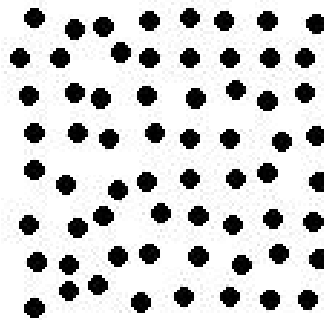
4. ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτό το κεφάλαιο διευκρινίζονται οι παράμετροι της υλοποίησης του particle swarm optimization algorithm και η μοντελοποίηση του προβλήματος. Επιπλέον, περιγράφονται τρία διαφορετικά σενάρια στα οποία εφαρμόστηκε ο αλγόριθμος.

4.1 Περιγραφή παραμέτρων προβλήματος

Στη βασική ιδέα του αλγορίθμου particle swarm optimization οι κόμβοι έχουν σταθερή ακτίνα, κατευθυνόμενη κεραία και καταναλώνουν ενέργεια κατά τη κίνηση τους προς τη βέλτιστη κατεύθυνση. Σε αυτή την υλοποίηση ακολουθείται μία διαφορετική μοντελοποίηση, διατηρώντας όμως αντιστοιχία με τους κανόνες του αρχικού αλγορίθμου. Οι κόμβοι είναι σταθεροί, αλλά για να μοντελοποιηθεί η κίνηση τους, όπως αυτή γίνεται στον αρχικό αλγόριθμο, έχουν τη δυνατότητα να διευρύνουν ή να συρρικνώνουν την εμβέλειά των κεραιών τους. Επιπλέον, οι κεραιές τους δεν θεωρούνται στην υλοποίηση κατευθυντικές.

Αρχικά οι κόμβοι έχουν μία περιορισμένη ποσότητα ενέργειας ($0,6 * SIZE$, όπου size είναι η διάσταση του πίνακα, δηλαδή $size * size$ είναι το πλήθος των particles) έτσι ώστε να μην είναι δυνατόν η πηγή να δίνει κατευθείαν τη πληροφορία σε όλους τους κόμβους. Η αρχική ακτίνα των particles είναι ίση με 1 και σε κάθε βήμα του αλγορίθμου ο κάθε κόμβος αυξάνει το πολύ κατά 1 την ακτίνα του ξοδεύοντας το ανάλογο ποσό ενέργειας. Οι κόμβοι σχηματίζουν ένα πλέγμα με μια μικρή χωρική διαταραχή όπως φαίνεται και στο Σχήμα 1.



Σχήμα 1: Διάταξη κόμβων

Η ποιότητα της πληροφορίας που μπορούν να έχουν οι κόμβοι κυμαίνεται στο διάστημα $[0,5]$. Αρχικά οι αναμεταδότες αρχικοποιούνται με τη βέλτιστη δυνατή τιμή ως προς τη ποιότητα της πληροφορίας. Σε κάθε βήμα του αλγορίθμου μειώνεται η ποιότητα της πληροφορίας που εκπέμπουν. Όταν αυτή φτάσει στο μηδέν, επανεκπέμπουν τη πληροφορία, καθώς αυτή θεωρείται απαρχαιωμένη. Στην ακόλουθη ενότητα περιγράφονται τα σενάρια υπό τα οποία εφαρμόστηκε ο αλγόριθμος.

4.2 Περιγραφή Σεναρίων

4.2.1 Σενάριο 1^ο: Απόκτηση πληροφορίας πλαισίου από όλους τους κόμβους

Στο σενάριο αυτό δεν εξετάζεται η επαναμετάδοση της πληροφορίας από τους πομπούς (transmitters) αφού άπαξ και πάρει τη πληροφορία ένας κόμβος σταματάει την αναζήτηση του. Επειδή υπάρχει περίπτωση να χάσει όλη του την ενέργεια κατά την

αναζήτηση, έχω εφαρμόσει 3 διαφορετικές πολιτικές αύξησης εμβέλειας (άρα και μείωσης ενέργειας) για τους κόμβους. Σε περίπτωση που ο κόμβος έχει πάνω από το 20% της ενέργειάς του, η αύξηση της εμβέλειας γίνεται όπως ακριβώς λέει ο αλγόριθμος PSO. Δηλαδή ως το $\max(0,x)$ όπου x είναι ένας τυχαίος αριθμός στο $[0,1]$. Αν έχει λιγότερο από 20% της ενέργειας τότε αυξάνουμε την εμβέλεια κατά $\max(0,x)/2$ και μειώνουμε την ενέργεια κατά το ίδιο. Τέλος, αν η ενέργεια είναι μικρότερη από 0,5 τότε δεν αυξάνουμε καθόλου την εμβέλεια και δεν χάνει άλλη ενέργεια ο κόμβος, περιμένοντας ουσιαστικά να λάβει τη πληροφορία από κάποιο γείτονα (δηλαδή κάποιο κόμβο που βρίσκεται εντός της εμβέλειας της ακτίνας του) ο οποίος έχει ακόμα αρκετή ενέργεια για να την αναζητήσει σε περιοχή που δεν φτάνει η εμβέλεια του κόμβου. Ο Πίνακας 5 συνοψίζει τα βήματα του αλγορίθμου για αυτό το σενάριο.

Πίνακας 5 : Ψευδοκώδικας περίπτωσης απόκτησης πληροφορίας πλαισίου από όλους τους κόμβους χωρίς περαιτέρω κριτήρια

1	Αρχικοποίηση μεγέθους του σμήνους s και άλλων παραμέτρων
2	Αρχικοποίηση των θέσεων των σωματιδίων με ψευδοτυχαίο τρόπο όπως δείχνει το Σχήμα 1 και της εμβέλειας των ακτίνων ίσης με τη μονάδα
3	Όσο δεν ισχύει το κριτήριο τερματισμού κάνε
4	$t=t+1$;
	Για κάθε κόμβο του σμήνους
5	Εάν ο κόμβος δεν έχει τη πληροφορία
6	Αύξησε την εμβέλεια της κεραίας του κατά ένα τυχαίο αριθμό $\epsilon \in [0, 1]$
7	Μείωσε την ενέργεια του κόμβου αναλόγως με τη πολιτική αύξησης εμβέλειας που ακολουθείται
8	Ενημέρωσε τους κόμβους που θεωρούνται γειτονιά
9	Υπολόγισε το fitness για το σωματίδιο όπως και στο κανονικό μοντέλο PSO
10	Ενημέρωσε το κόμβο με τη καλύτερη ποιοτικά πληροφορία από τη γειτονιά
10	Τέλος εάν
11	Επόμενος κόμβος
12	Τέλος του όσο

4.2.2 Σενάριο 2^ο: Απόκτηση πληροφορίας πλαισίου από τους πλησιέστερους γείτονες

Σε αυτό το σενάριο ο κάθε κόμβος προσπαθεί να βελτιστοποιήσει την ακτίνα στην οποία βρήκε τη πληροφορία. Για να το πετύχει αυτό σε κάθε βήμα αυξάνει την εμβέλεια της ακτίνας του μέχρι να βρει τη πληροφορία. Μόλις τη βρει, αρχίζει να μειώνει σταδιακά την εμβέλεια για να βρει τη πληροφορία πιο κοντά. Αν ο κόμβος βρει πιο κοντά τη πληροφορία τότε μειώνεται η βέλτιστη ακτίνα. Αν όμως, καθώς μειώνει την ακτίνα δεν υπάρχει πλέον κάποιον γείτονα να του δώσει την πληροφορία, τότε ο κόμβος αρχίζει πάλι να ανοίγει την εμβέλεια της ακτίνας του με σκοπό να ξαναβρεί τη πληροφορία.

Επιπλέον μετρική σε αυτό το σενάριο αποτελεί η χρονική ποιότητα πληροφορίας που παίρνει ο κάθε κόμβος. Κάθε κόμβος ελέγχει σε κάθε βήμα όλους του γείτονες του και προτιμάει αυτόν που του δίνει τη πιο ποιοτική χρονικά πληροφορία. Ο Πίνακας 6 παρουσιάζει συνοπτικά τα βήματα του αλγορίθμου για το συγκεκριμένο σενάριο.

Πίνακας 6 : Ψευδοκώδικας περίπτωσης απόκτησης πληροφορίας πλαισίου από τον πλησιέστερο γείτονα

1	Αρχικοποίηση μεγέθους του σμήνους s και άλλων παραμέτρων
2	Αρχικοποίηση των θέσεων των σωματιδίων με ψευδοτυχαίο τρόπο όπως δείχνει το Σχήμα 1 και της εμβέλειας των ακτίνων ίσης με τη μονάδα
3	Όσο δεν ισχύει το κριτήριο τερματισμού κάνε
4	$t=t+1$;
5	Για κάθε κόμβο i του σμήνους που έχει τη πληροφορία
6	Μείωσε τη ποιότητα της πληροφορίας που έχει ο κόμβος
7	Εάν ο κόμβος είναι transmitter και η ποιότητα της πληροφορίας είναι 0
8	Επαναμετάδωσε τη πληροφορία
9	Τέλος του εάν
10	Επόμενος κόμβος i
11	Για κάθε κόμβο i του σμήνους
12	Εάν ο κόμβος δεν έχει τη πληροφορία
13	Αύξησε την εμβέλεια της κεραίας του κατά ένα τυχαίο αριθμό $\epsilon [0, 1]$
14	Μείωσε την ενέργεια του κόμβου αναλόγως με τη πολιτική αύξησης εμβέλειας που ακολουθείται
15	Ενημέρωσε τους κόμβους που θεωρούνται γειτονιά
16	Υπολόγισε το fitness για το σωματίδιο όπως και στο κανονικό μοντέλο PSO. Σε αυτό το σενάριο ως \mathbf{x}^* θα επιλεγεί ο κοντινότερος γείτονας με τη πληροφορία.
17	$\mathbf{x}^* = \operatorname{argmin}_{i=1}^n (f(\mathbf{x}^*(t-1)), f(\mathbf{x}_1(t)), f(\mathbf{x}_2(t)), \dots, f(\mathbf{x}_i(t)), \dots, f(\mathbf{x}_n(t)))$;
18	Αλλιώς
19	Μείωσε την ακτίνα ώστε να βρείς τη πληροφορία όσο το δυνατόν πιο τοπικά
20	Επόμενος κόμβος i
21	Τέλος του όσο

4.2.3 Σενάριο 3^ο: Απόκτηση κατά το δυνατόν ποιοτικότερης χρονικά πληροφορίας από το κάθε κόμβο

Σε αυτό το σενάριο ο κάθε κόμβος ελέγχει σε κάθε βήμα αν κάποιος από τους γείτονές του έχει τη πληροφορία. Μόλις τη βρει σε κάποιον γείτονα τότε σταματάει την αναζήτηση. Η ποιότητα της πληροφορίας που αποκτά ο κάθε κόμβος είναι αντιστρόφως

ανάλογη του χρόνου και ο κάθε κόμβος ξεκινάει πάλι να αναζητά πληροφορία όταν αυτή που έχει θεωρηθεί απαρχαιωμένη, ή η ποιότητά της πέσει στο μηδέν. Οι transmitters επαναμεταδίδουν τη πληροφορία όταν η ποιότητά της φτάσει στο μηδέν. Στο Πίνακα 7 φαίνονται συνοπτικά τα βήματα του αλγορίθμου στη περίπτωση που η πολιτική που ακολουθείται επιβάλλει ο κάθε κόμβος να αναζητά κατά το δυνατόν ποιοτικότερη χρονικά πληροφορία πλαισίου στη γειτονία του.

Πίνακας 7 : Ψευδοκώδικας περίπτωσης απόκτησης κατά το δυνατόν ποιοτικότερης χρονικά πληροφορίας πλαισίου

1	Αρχικοποίηση μεγέθους του σμήνους s και άλλων παραμέτρων
2	Αρχικοποίηση των θέσεων των σωματιδίων με ψευδοτυχαίο τρόπο όπως δείχνει το Σχήμα 1 και της εμβέλειας των ακτίνων ίσης με τη μονάδα
3	Όσο δεν ισχύει το κριτήριο τερματισμού κάνε
4	$t=t+1$;
5	Για κάθε κόμβο i του σμήνους που έχει τη πληροφορία
6	Μείωσε τη ποιότητα της πληροφορίας που έχει ο κόμβος
7	Εάν ο κόμβος είναι transmitter και η ποιότητα της πληροφορίας είναι 0
8	Επαναμετάδωσε τη πληροφορία
9	Τέλος του εάν
10	Επόμενος κόμβος i
11	Για κάθε κόμβο i του σμήνους
12	Εάν ο κόμβος δεν έχει τη πληροφορία
13	Αύξησε την εμβέλεια της κεραίας του κατά ένα τυχαίο αριθμό $\epsilon [0, 1]$
14	Μείωσε την ενέργεια του κόμβου αναλόγως με τη πολιτική αύξησης εμβέλειας που ακολουθείται
15	Ενημέρωσε τους κόμβους που θεωρούνται γειτονιά
16	Υπολόγισε το fitness για το σωματίδιο όπως και στο κανονικό μοντέλο PSO. Σε αυτό το σενάριο ως \mathbf{x}^* θα επιλέγει ο γείτονας με τη καλύτερη χρονικά πληροφορία.
17	$\mathbf{x}^* = \operatorname{argmin}_{i=1}^n (f(\mathbf{x}^*(t-1)), f(\mathbf{x}_1(t)), f(\mathbf{x}_2(t)), \dots, f(\mathbf{x}_i(t)), \dots, f(\mathbf{x}_n(t)))$;
18	Αλλιώς
19	Αν διαθέτεις τη πληροφορία για ένα χρονικό διάστημα μεγαλύτερο από ένα κατώτατο όριο (delay_time) ή αν η ποιότητα της πληροφορίας είναι 0
20	Θεώρησε ότι ο κόμβος δεν έχει πλέον τη πληροφορία
21	Επόμενος κόμβος i
22	Τέλος του όσο

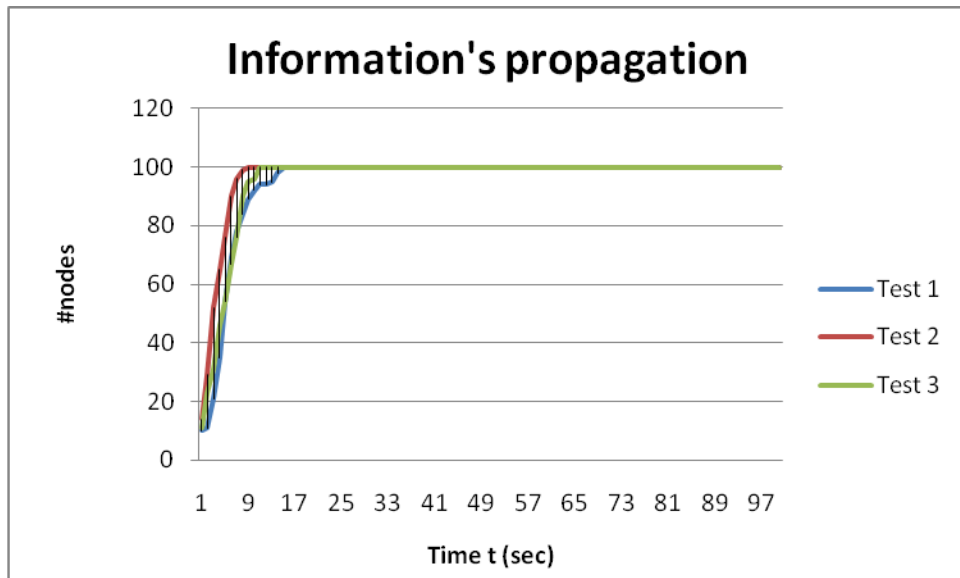
5. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Στο κεφάλαιο αυτό παρουσιάζονται τα πειραματικά αποτελέσματα από την εκτέλεση όλων των σεναρίων που αναλύθηκαν στη προηγούμενη ενότητα. Σε όλα τα σεναρία παίρνουμε μετρήσεις για την κατανάλωση ενέργειας του συστήματος και το πλήθος των κόμβων που λαμβάνουν τη πληροφορία. Στα σεναρία 2 και 3 μελετάται και η μέση βέλτιστη ακτίνα των κόμβων. Η αρχική τιμή που έχει δοθεί στη βέλτιστη ακτίνα των κόμβων θα έπρεπε να είναι άπειρη (όπως λέει και ο αλγόριθμος). Επειδή αυτό είναι ανέφικτο, η μεταβλητή αυτή έχει ως αρχική τιμή μία τιμή την οποία δεν θα φτάσει ποτέ η ακτίνα της κεραίας του κόμβου (λόγω της περιορισμένης ενέργειας που έχει).

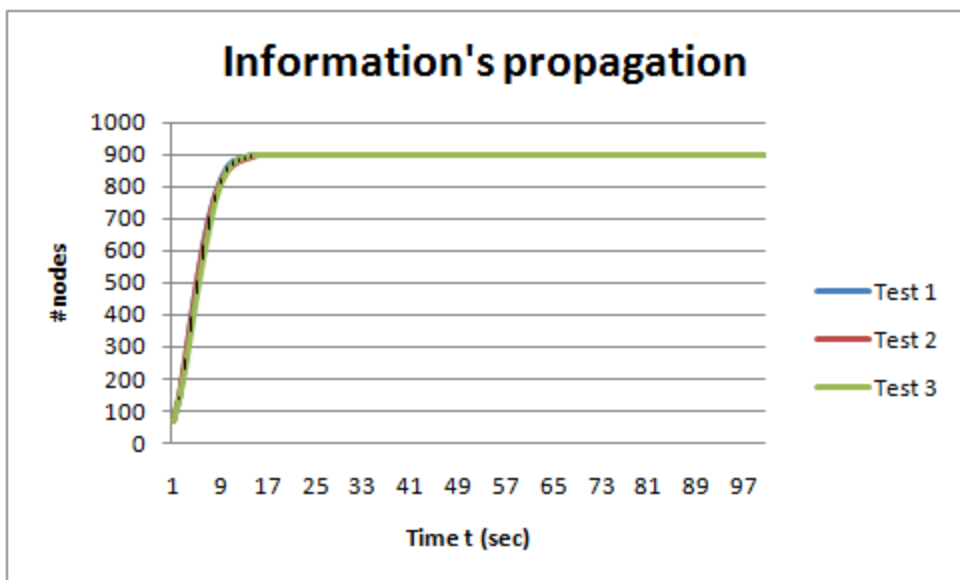
Για κάθε σενάριο παρουσιάζονται δύο σειρές πειραμάτων. Στη πρώτη σειρά πειραμάτων ακολουθείται η ίδια πολιτική ως προς την αύξηση της εμβέλειας της κεραίας, άρα και τη κατανάλωση της ενέργειας, για όλους τους κόμβους ανεξάρτητα από την εναπομείνουσα ενέργεια του καθενός. Στη δεύτερη σειρά πειραμάτων ακολουθείται μία εναλλακτική προσέγγιση ως προς την αύξηση της εμβέλειας της κεραίας του κάθε κόμβου, με βάση την διαθέσιμη ενέργεια. Πιο συγκεκριμένα, αν ο κόμβος έχει περισσότερο από το 20% της αρχικής του ενέργειας τότε, διευρύνει κανονικά την εμβέλεια της κεραίας του, όπως αποφάσισε. Αλλιώς, αν έχει περισσότερο από ένα ελάχιστο όριο, αλλά λιγότερο από το 20% της αρχικής του ενέργειας, τότε αποφασίζει να διευρύνει κατά το ήμισυ την εμβέλεια της ακτίνας του σε σχέση με την αρχική του απόφαση. Αν έχει λιγότερη ενέργεια και από το κατώτατο όριο, τότε δεν διευρύνει καθόλου την εμβέλεια της κεραίας του και αναμένει λόγω της συνεργατικής συμπεριφοράς του συστήματος, να αποκτήσει στο μέλλον την πληροφορία από κάποιο κόμβο που βρίσκεται ήδη στους γείτονες του, αλλά δεν έχει τη τρέχουσα χρονική στιγμή τη πληροφορία πλαισίου.

5.1 Αποτελέσματα προσομοίωσης για το 1^ο σενάριο

Το Σχήμα 2 και το Σχήμα 3 δείχνουν τη διάδοση της πληροφορίας σε περιπτώσεις που το δίκτυο αποτελείται από 100 και από 900 κόμβους αντίστοιχα. Όπως φαίνεται και από τα δύο σχήματα, οι κόμβοι αποκτούν τη πληροφορία με εκθετικό ρυθμό, ενώ η διάδοση της πληροφορίας είναι ανεξάρτητη από το πλήθος των κόμβων. Η συμπεριφορά αυτή είναι αναμενόμενη, με βάση το σενάριο, αφού κάθε κόμβος σταματάει την περαιτέρω αναζήτηση άπαξ και αποκτήσει τη πληροφορία πλαισίου.

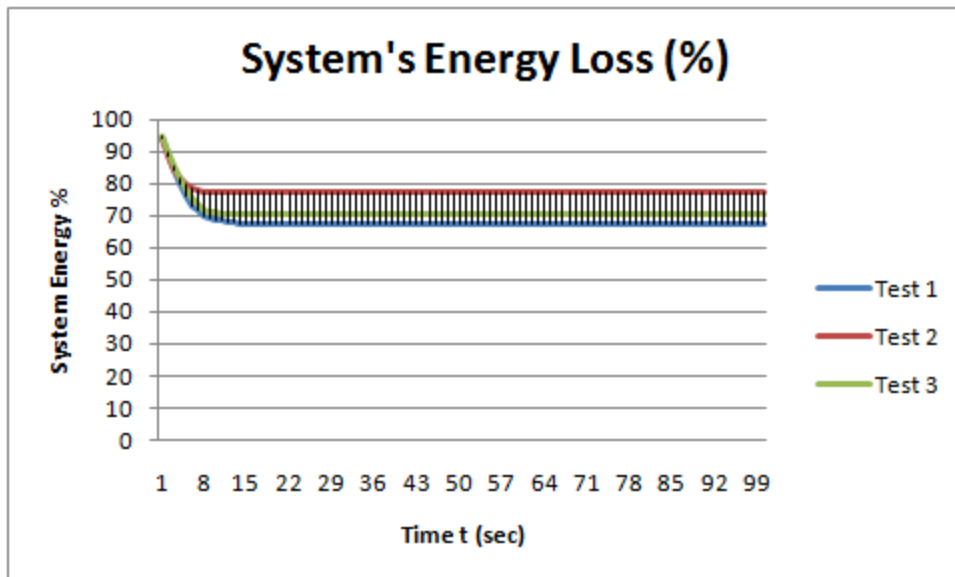


Σχήμα 2: Διάδοση πληροφορίας σε δίκτυο με 100 κόμβους

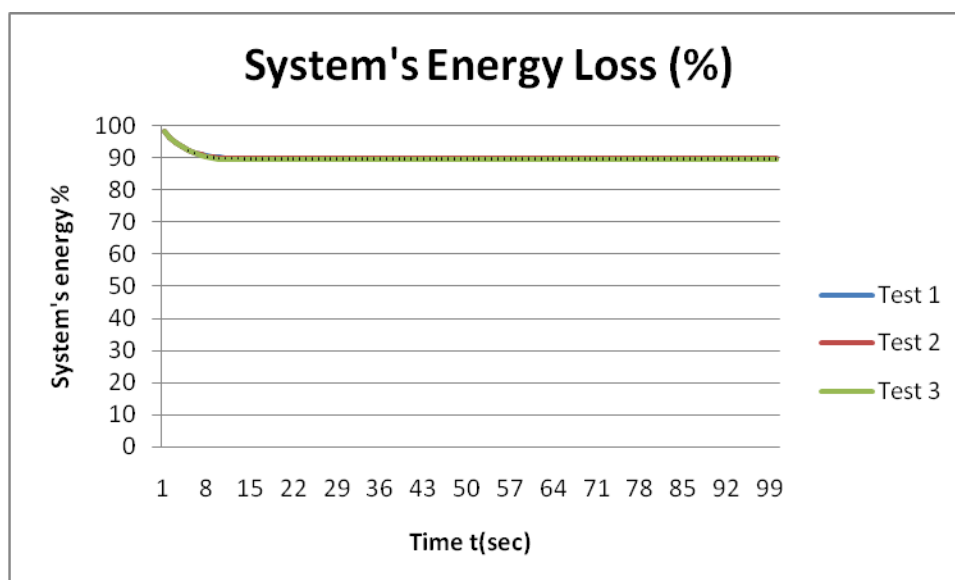


Σχήμα 3: Διάδοση πληροφορίας σε δίκτυο με 900 κόμβους

Το Σχήμα 4 και το Σχήμα 5 δείχνουν το ποσοστό επί τοις εκατό της απώλειας ενέργειας του συστήματος. Το πείραμα αυτό, αντικατοπτρίζει το ποσοστό ενέργειας που καταναλώνουν οι κόμβοι κατά τη διαδικασία μεταβολής της εμβέλειας των κεραιών τους με σκοπό την ανάκτηση της πληροφορίας πλαισίου.

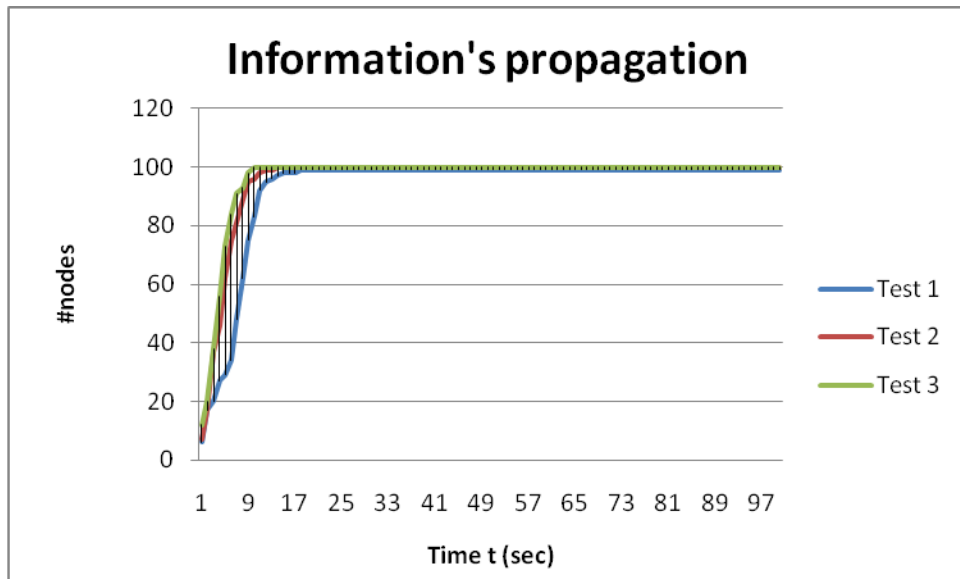


Σχήμα 4: Απώλεια ενέργειας συστήματος για δίκτυο με 100 κόμβους

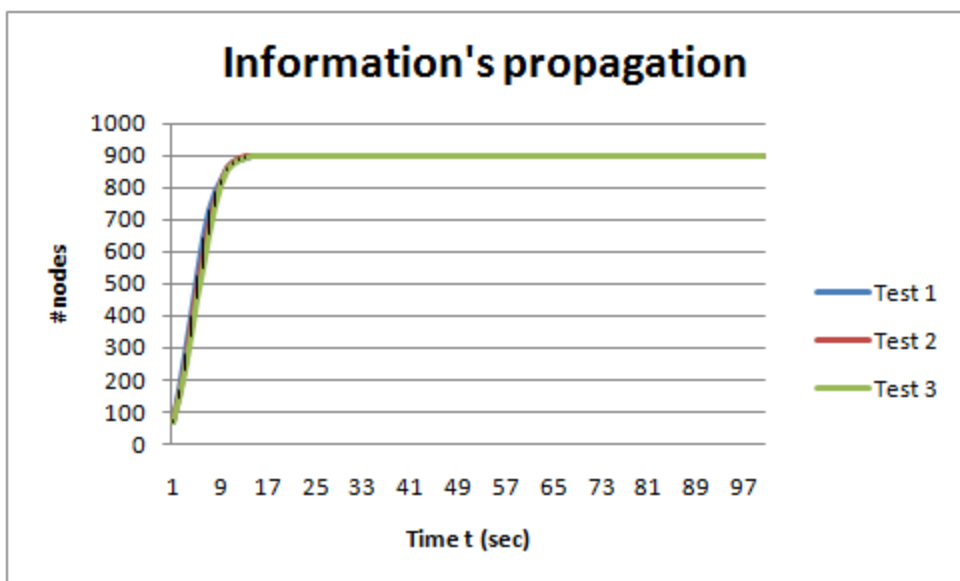


Σχήμα 5: Απώλεια ενέργειας συστήματος για δίκτυο με 900 κόμβους

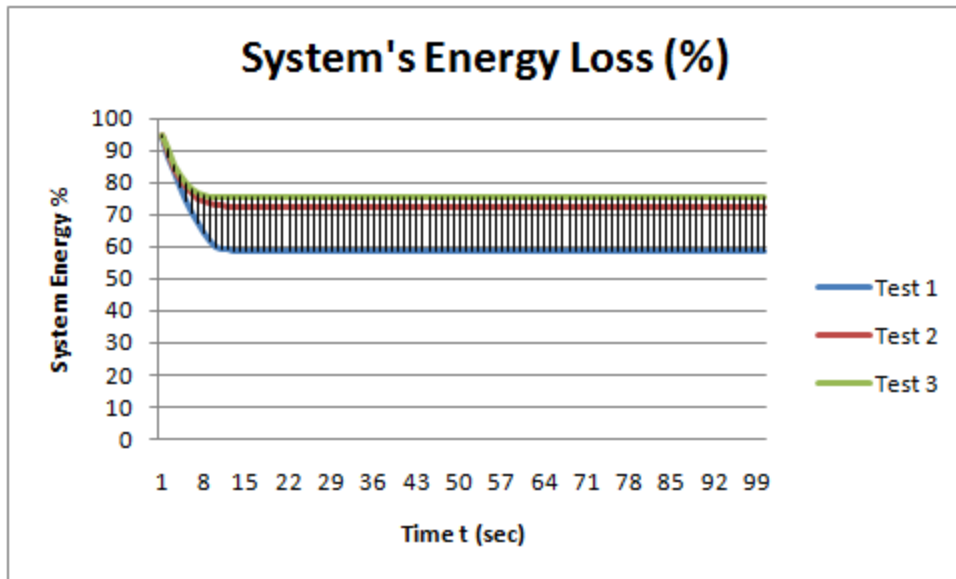
Στο παραπάνω πείραμα σε κάθε βήμα, ο κάθε κόμβος ελέγχει αν έχει την πληροφορία πλαισίου. Αν όχι, τότε διευρύνει την εμβέλεια της κεραίας του, ώστε να βρει τη πληροφορία σε πιο απομακρυσμένους από αυτόν κόμβους. Ακολουθώντας, στο Σχήμα 6, Σχήμα 7, Σχήμα 8 και Σχήμα 9 παρουσιάζονται τα αποτελέσματα της δεύτερης σειράς πειραμάτων, όπου ο κάθε κόμβος διευρύνει την ακτίνα της κεραίας του με βάση τη διαθέσιμη ενέργεια. Όπως φαίνεται και από τα σχήματα, δεν υπάρχει μεγάλη απόκλιση σε σχέση με τη πρώτη σειρά πειραμάτων. Αυτό οφείλεται στο γεγονός ότι οι κόμβοι αποκτούν πολύ γρήγορα τη πληροφορία, πριν προλάβουν να καταναλώσουν πολύ ενέργεια, με αποτέλεσμα, εν τέλει οι περισσότεροι, αν όχι όλοι, να ακολουθούν την ίδια πολιτική ως προς την διεύρυνση της ακτίνας της κεραίας τους.



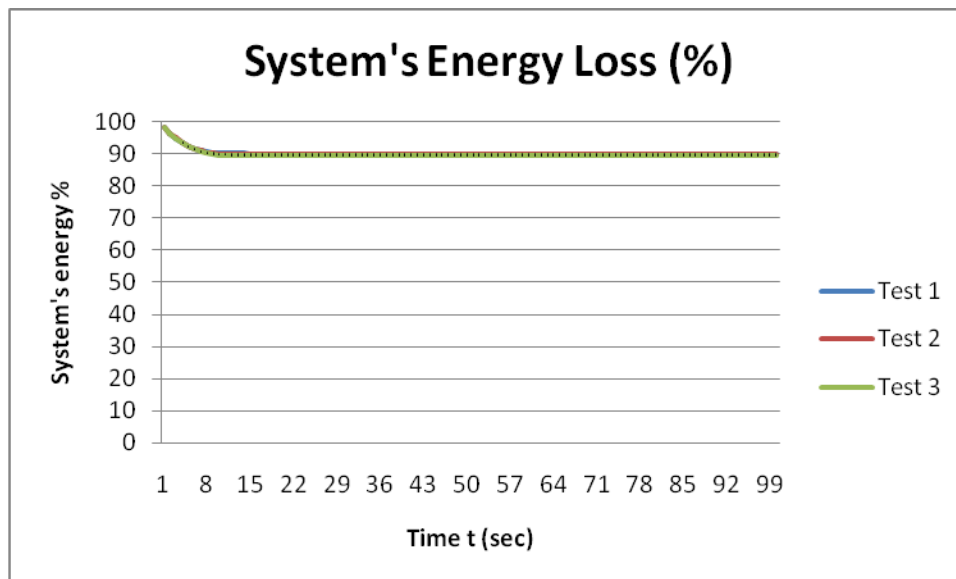
Σχήμα 6: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους



Σχήμα 7: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους



Σχήμα 8: Απώλεια ενέργειας συστήματος για δίκτυο με 100 κόμβους



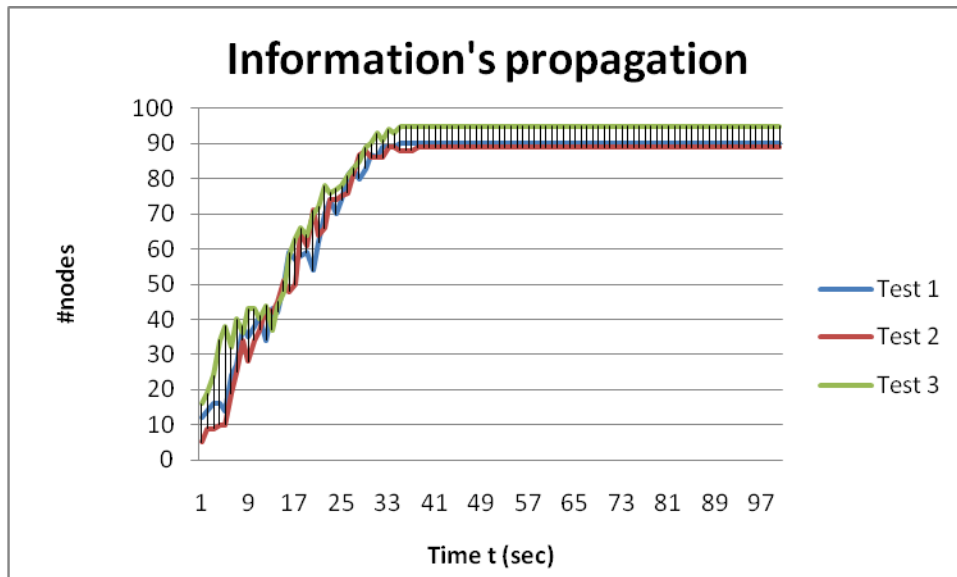
Σχήμα 9: Απώλεια ενέργειας συστήματος για δίκτυο με 900 κόμβους

5.2 Αποτελέσματα προσομοίωσης για το 2^ο σενάριο

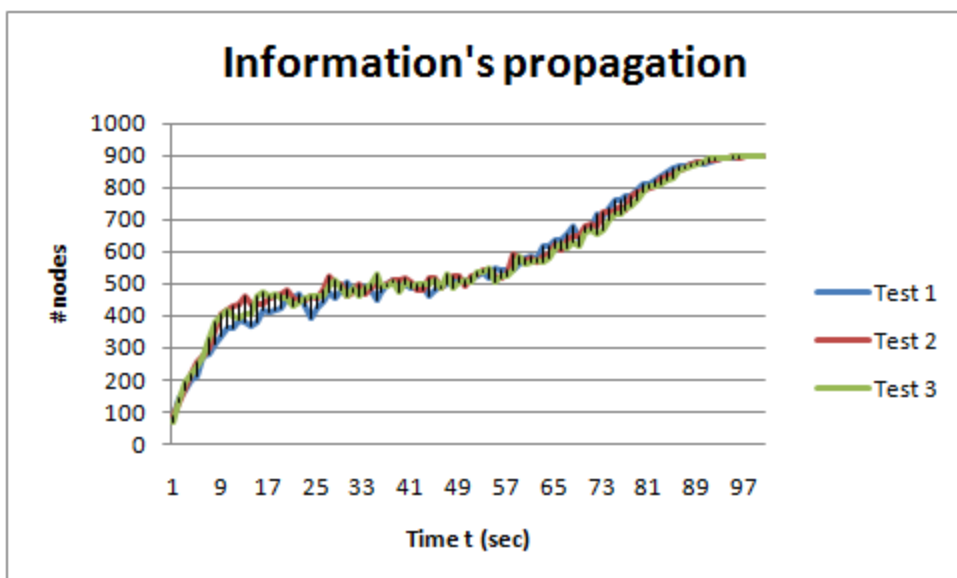
Στο 2^ο σενάριο, όπως αναλύθηκε και στην ενότητα 4.2.2 ο κάθε κόμβος αφού αποκτήσει τη πληροφορία, προσπαθεί να βελτιστοποιήσει την ακτίνα του, ώστε να αποκτή τη πληροφορία πλαισίου, από κατά το δυνατόν κοντινότερο γείτονα και να για καταναλώνει λιγότερη ενέργεια στο μέλλον. Η χρονική ποιότητα της πληροφορίας διαδραματίζει δευτερεύων ρόλο σε αυτό το σενάριο, αφού λαμβάνεται μεν υπόψη, αλλά μόνο σε περιπτώσεις όπου ο κόμβος μπορεί να αποκτήσει τη πληροφορία από περισσότερους

του ενός κόμβους. Τότε, επιλέγει να πάρει τη πληροφορία από το γείτονα με την ποιοτικότερη χρονικά πληροφορία.

Όπως φαίνεται και από το Σχήμα 10 και το Σχήμα 11 η καμπύλη ανάκτησης της πληροφορίας είναι μικρότερη, το οποίο υποδεικνύει ότι οι κόμβοι χρειάζονται περισσότερο χρόνο σε σχέση με το 1^ο σενάριο, για να αποκτήσουν την πληροφορία. Μία εξίσου βασική διαφορά με το 1^ο σενάριο είναι η εξάρτηση που παρατηρείται σε σχέση με το πληθυσμό των particles. Πιο συγκεκριμένα, αύξηση του πληθυσμού, προκαλεί επιβράδυνση στη διάδοση της πληροφορίας.

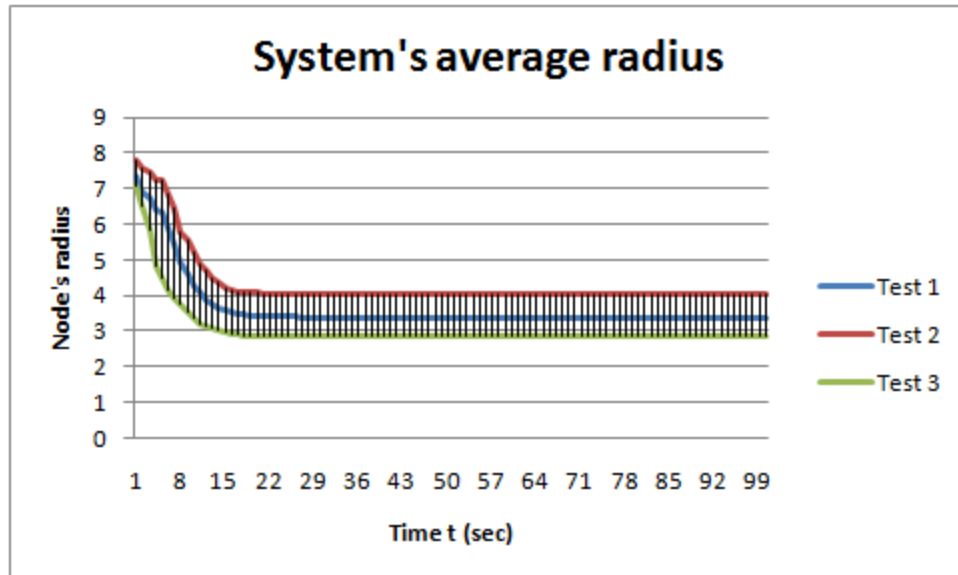


Σχήμα 10: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους

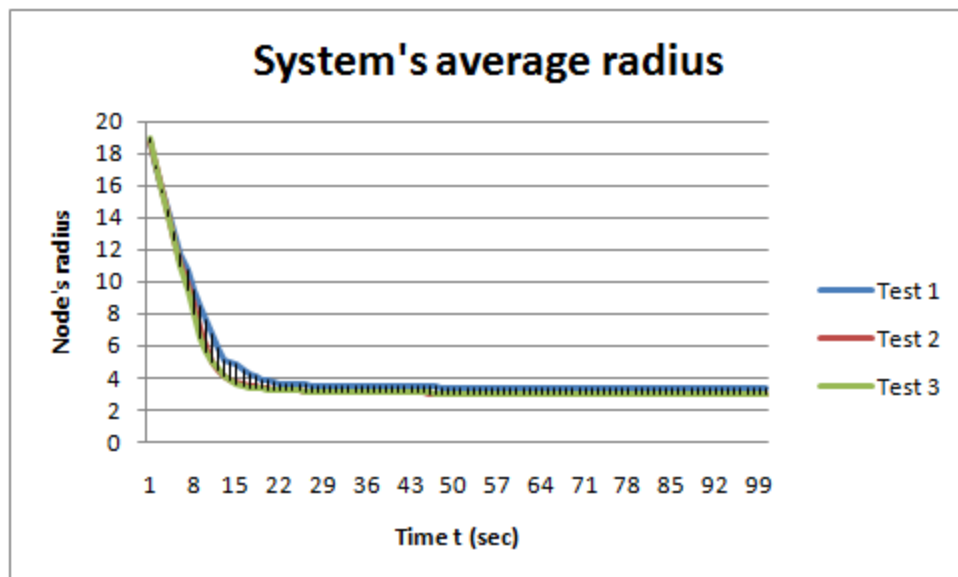


Σχήμα 11: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους

Το Σχήμα 12 και το Σχήμα 13 δείχνουν τη μείωση της μέσης ακτίνας σε περίπτωση 100 και 900 κόμβων αντίστοιχα. Η μείωση αυτή προκαλείται από τη προσπάθεια των κόμβων να βρουν τη πληροφορία πλαισίου όσο πιο τοπικά γίνεται. Όπως φαίνεται και από τα δύο αυτά σχήματα, το σύστημα είναι scalable δηλαδή, παρουσιάζει καλύτερα αποτελέσματα καθώς αυξάνεται το πλήθος των κόμβων.

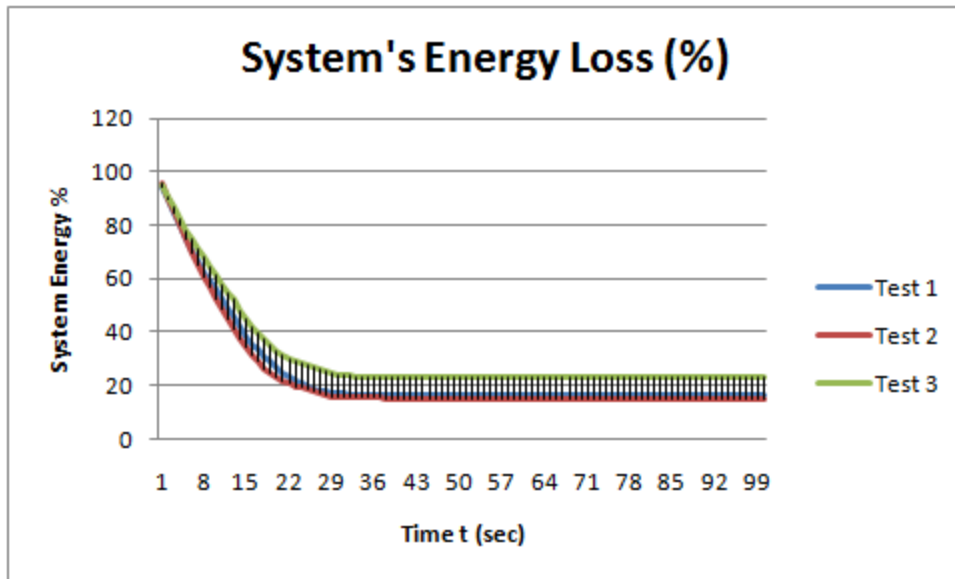


Σχήμα 12: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους

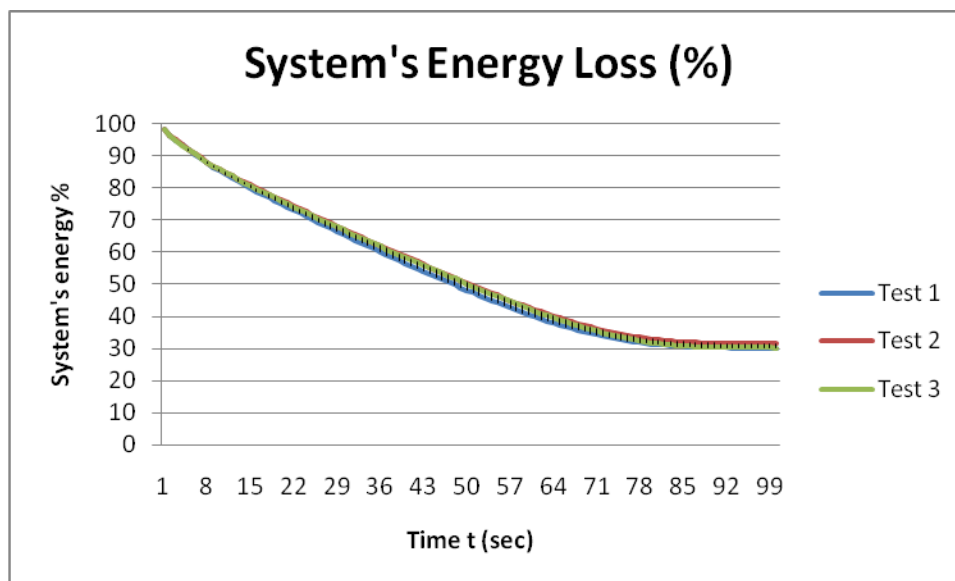


Σχήμα 13: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους

Η μείωση της μέσης ακτίνας των κόμβων του δικτύου, οδηγεί σε μειωμένη κατανάλωση ενέργειας από το σύστημα (Σχήμα 14 και Σχήμα 15). Έτσι, όταν το σύστημα αποτελείται από 900 κόμβους παρατηρείται εξοικονόμηση ενέργειας της τάξης του 10% σε σχέση με το σύστημα που αποτελείται από 100 κόμβους.

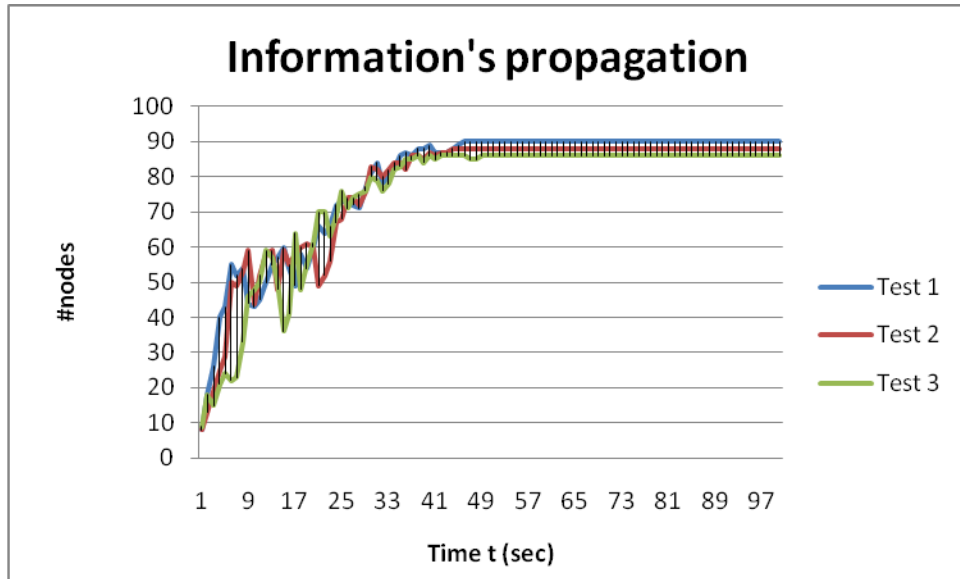


Σχήμα 14: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους

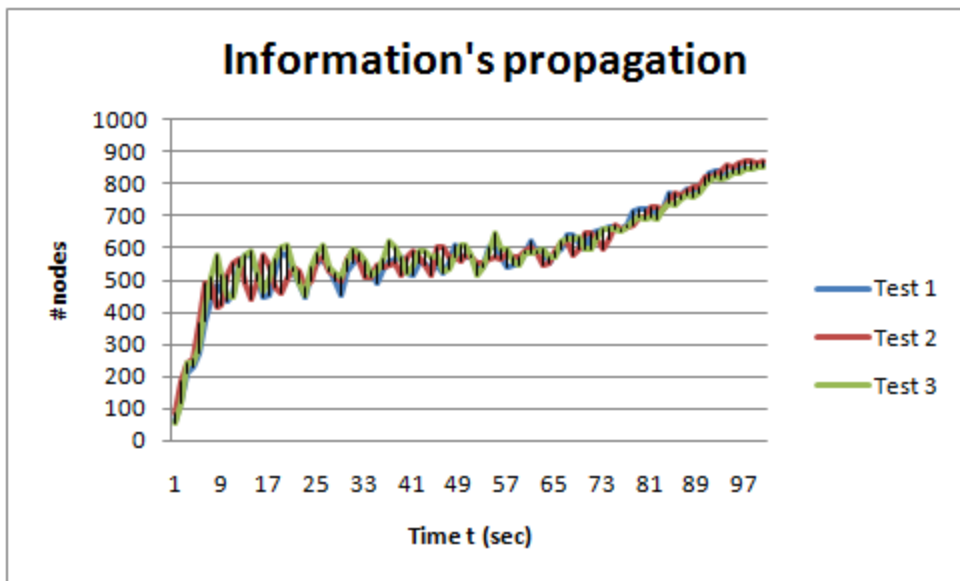


Σχήμα 15: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους

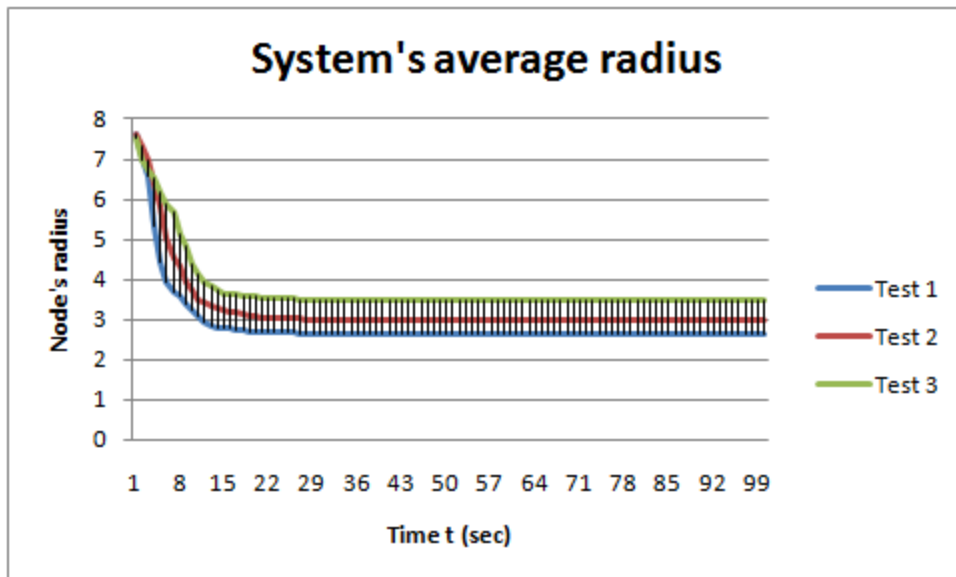
Στα ακόλουθα σχήματα παρουσιάζονται τα πειραματικά αποτελέσματα ως προς τη διάδοση της πληροφορίας πλαισίου, τη μέση ακτίνα και την μέση απώλεια ενέργειας των κόμβων του συστήματος, αν ακολουθηθεί διαφορετική πολιτική ως προς την αυξομείωση της εμβέλειας των ακτινών των κόμβων με βάση την διαθέσιμη ενέργειά τους. Τα πειραματικά αποτελέσματα δείχνουν ότι, ενώ η διάδοση της πληροφορίας ακολουθεί τον ίδιο ρυθμό (Σχήμα 16 και Σχήμα 17) σε σχέση με την ενιαία πολιτική (Σχήμα 10 και Σχήμα 11), επιτυγχάνεται σημαντική μείωση ως προς τη κατανάλωση ενέργειας των κόμβων (Σχήμα 20 και Σχήμα 21), ενώ παρατηρείται και ανεπαισθητη μείωση στην μέση ακτίνα των κόμβων του συστήματος (Σχήμα 18 και Σχήμα 19).



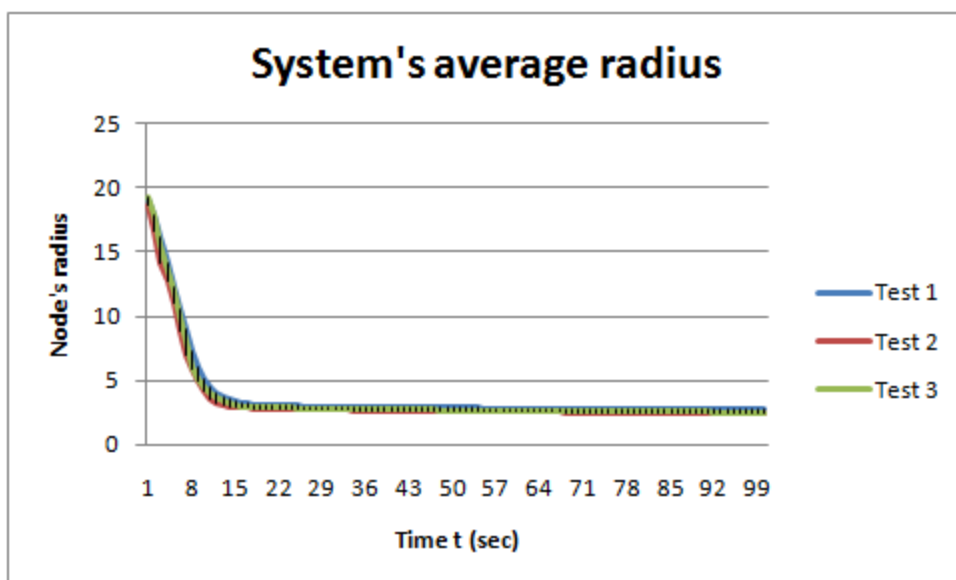
Σχήμα 16: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους



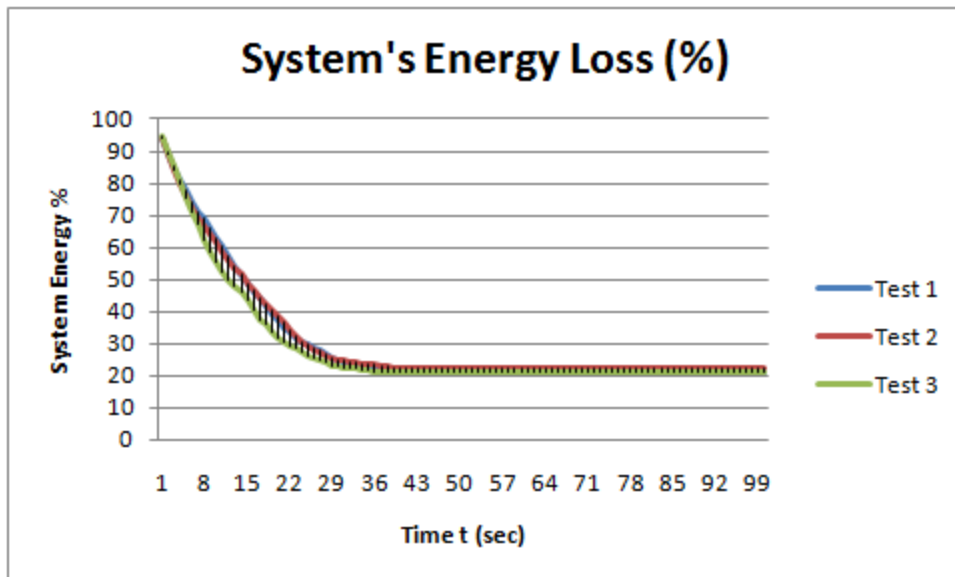
Σχήμα 17: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους



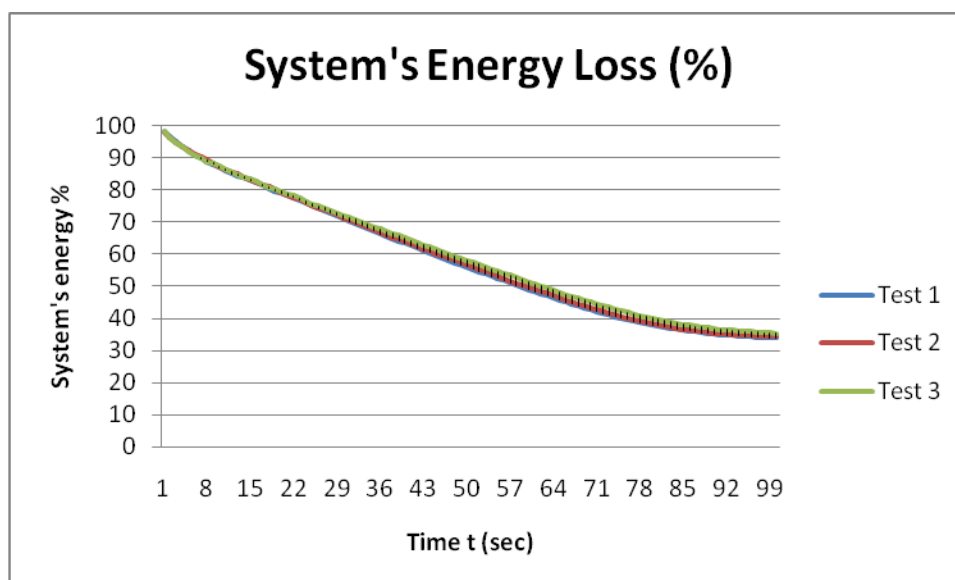
Σχήμα 18: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους



Σχήμα 19: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους



Σχήμα 20: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους

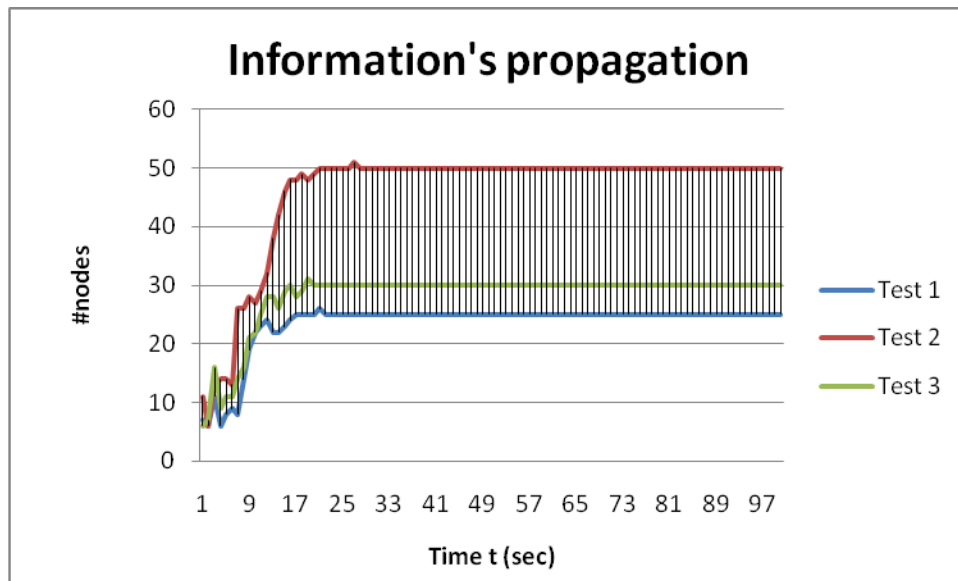


Σχήμα 21: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους

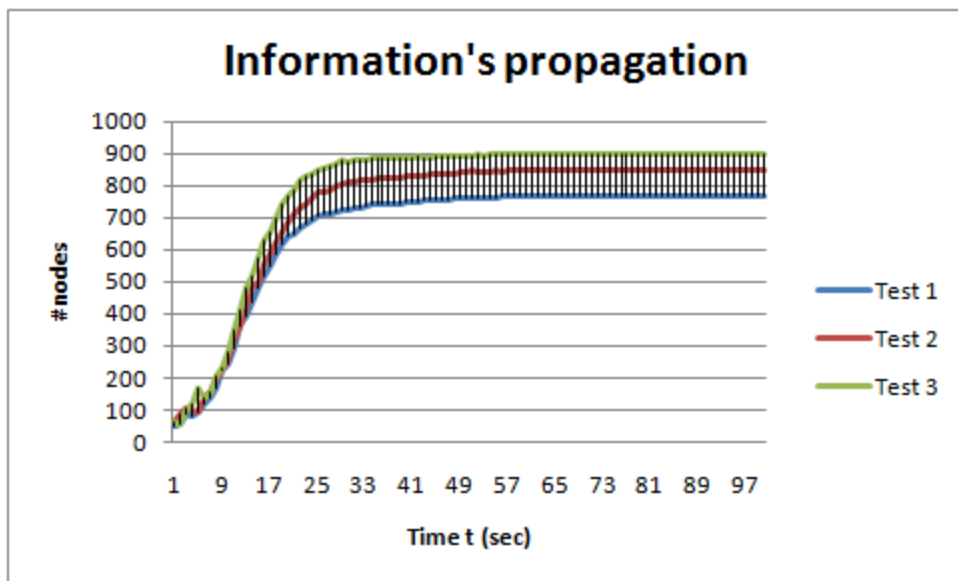
5.3 Αποτελέσματα προσομοίωσης για το 3^ο σενάριο

Στο 3^ο σενάριο, όπως αναλύθηκε και στην ενότητα 4.2.3 η χρονική ποιότητα της πληροφορίας διαδραματίζει κύριο ρόλο. Ο κάθε κόμβος αφού αποκτήσει τη πληροφορία, παύει την αναζήτηση και την επανεκκινεί μόνο αφού θεωρήσει τη πληροφορία που απέκτησε απαρχαιωμένη. Το πρόβλημα σε αυτό το σενάριο είναι δυναμικής φύσεως, αφού με το πέρασμα του χρόνου το πλαίσιο που αποκτούν οι κόμβοι θεωρείται απαρχαιωμένο και εξαναγκάζονται να ξανανοήξουν τη κεραία τους ώστε να αναζητήσουν πάλι τη πληροφορία πλαισίου. Το Σχήμα 22 δείχνει ότι για μικρό σύστημα ενδέχεται να υπάρχουν μεγάλες αποκλίσεις στο πλήθος των κόμβων που θα

καταφέρουν να αποκτήσουν τη πληροφορία πλαισίου, ενώ συγκρίνοντας το με το Σχήμα 23 η απόκλιση ανάμεσα στα πειράματα είναι μικρότερη για τους 900 κόμβους.

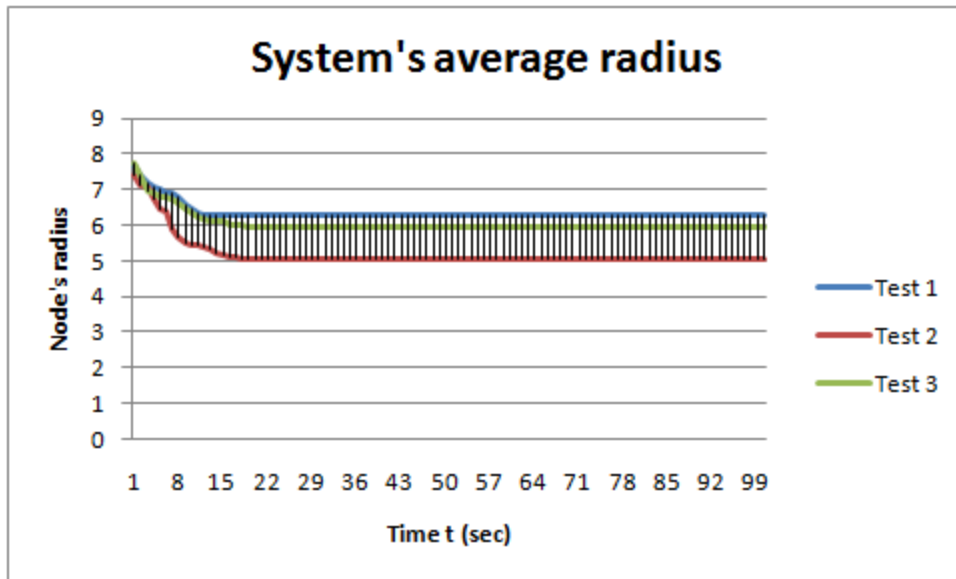


Σχήμα 22: Διάδοση πληροφορίας για δίκτυο με 100 κόμβους

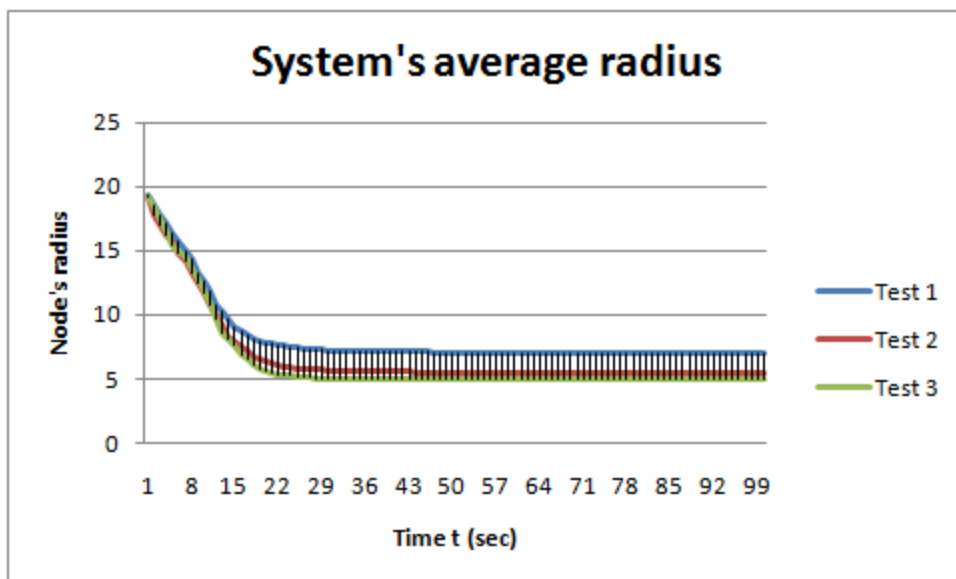


Σχήμα 23: Διάδοση πληροφορίας για δίκτυο με 900 κόμβους

Το Σχήμα 24 και το Σχήμα 25 παρουσιάζουν τη μέση ακτίνα των κόμβων του συστήματος. Σε αυτό το σενάριο αποτελεί δευτερεύων παράγοντα η βελτιστοποίηση αυτής της μετρικής, παρόλα αυτά στις περιπτώσεις που το particle μπορεί να πάρει ίδιας χρονικής ποιότητας πληροφορία από δύο ή περισσότερα γειτονικά particles, προτιμάει το πλησιέστερο.

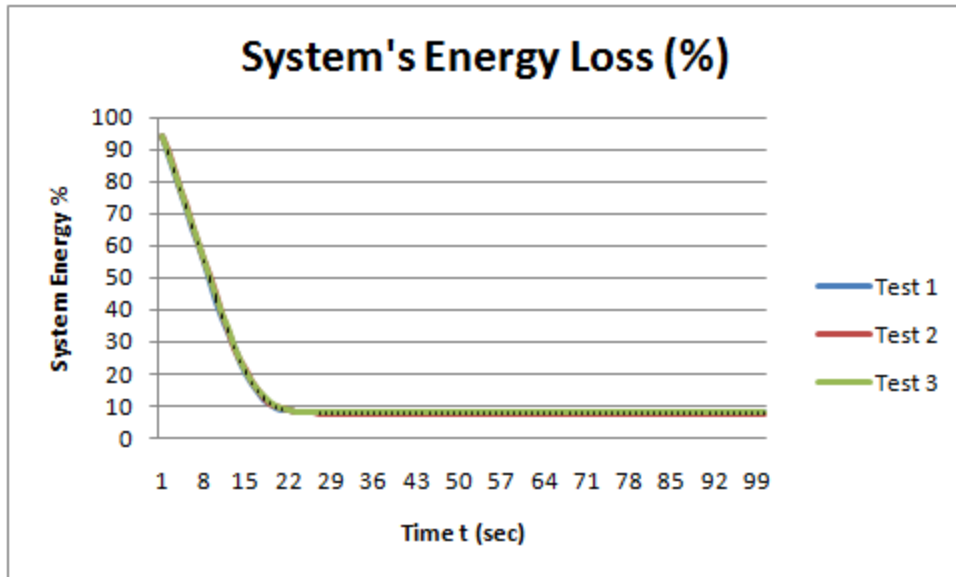


Σχήμα 24: Μέση ακτίνα συστήματος για δίκτυο με 100 κόμβους

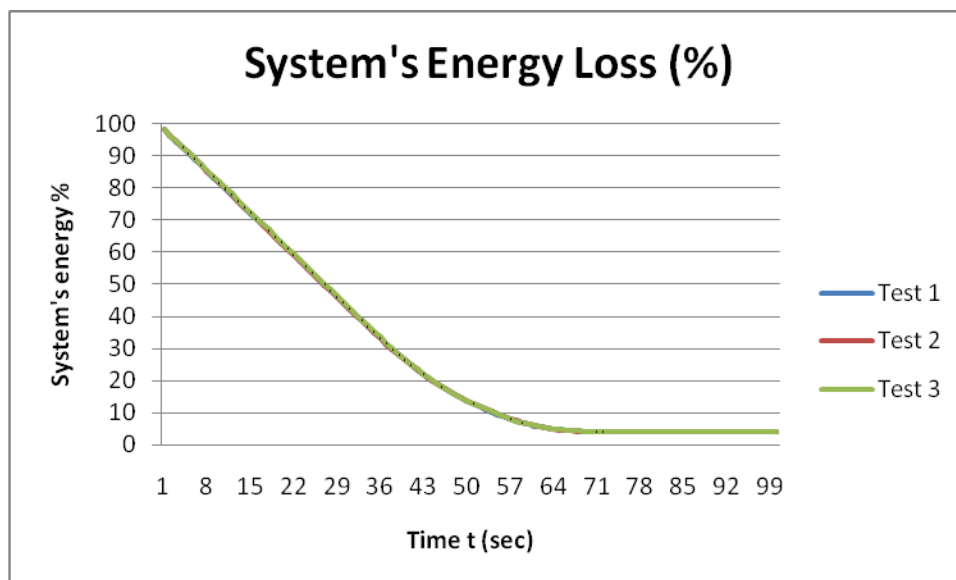


Σχήμα 25: Μέση ακτίνα συστήματος για δίκτυο με 900 κόμβους

Τέλος, στο Σχήμα 26 και στο Σχήμα 27 φαίνεται η απώλεια ενέργειας του συστήματος. Όπως φαίνεται, το σύστημα οδηγείται σχεδόν σε ολική απώλεια της ενέργειας. Αυτή η συμπεριφορά του συστήματος είναι απόλυτα φυσιολογική, καθώς οι κόμβοι δεν σταματούν ποτέ να αναζητούν πληροφορία.



Σχήμα 26: Μέση απώλεια συστήματος για δίκτυο με 100 κόμβους



Σχήμα 27: Μέση απώλεια συστήματος για δίκτυο με 900 κόμβους

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα πτυχιακή εργασία μελετήθηκαν τρία πιθανά σενάρια για ένα σύστημα κόμβων που προσπαθούν να ανακαλύψουν πληροφορία πλαισίου και μοντελοποιούνται με βάση τον αλγόριθμο Particle Swarm Optimization. Σε κάθε ένα από αυτά τα σενάρια, δίνεται διαφορετική βαρύτητα στις παραμέτρους του προβλήματος, ανάλογα με το βασικό στόχο του σεναρίου.

Το σενάριο 1 αναδεικνύει τη γρήγορη σύγκλιση του αλγορίθμου και τη μικρή κατανάλωση ενέργειας του συστήματος όταν δεν εξετάζονται επιπλέον βελτιστοποιήσεις, όπως:

- απόκτηση πληροφορίας από όσο το δυνατόν πιο κοντινό γειτονικό κόμβο
- χρονική εγκυρότητα της πληροφορίας πλαισίου

Το σενάριο 2 προσπαθεί κυρίως να βελτιώσει την ακτίνα της κεραίας του κάθε κόμβου, ώστε μελλοντικά να καταναλώνει λιγότερη ενέργεια. Κατά δεύτερο ρόλο μας ενδιαφέρει να αποκτήσει και όσο το δυνατόν χρονικά έγκυρη πληροφορία, αλλά μόνο εφόσον δύναται να αποκτήσει τη πληροφορία από περισσότερους του ενός κόμβους.

Στο σενάριο 3 παρουσιάστηκε μία προσέγγιση, όπου ο κάθε κόμβος ενδιαφέρεται να έχει χρονικά έγκυρη πληροφορία. Η ποιότητα της πληροφορίας φθίνει καθώς περνάει ο χρόνος και οι πομποί επαναμεταδίδουν την πληροφορία που έχουν όταν η ποιότητά της φτάσει στο μηδέν.

Τέλος, πέρα από τα 3 σενάρια, ακολουθήθηκαν δύο διαφορετικές πολιτικές ως προς την αύξηση της ακτίνας της κεραίας του κάθε κόμβου. Στη μία πολιτική οι κόμβοι ακολουθούν ενιαία πολιτική ως προς την αύξηση της εμβέλειας της κεραίας, ενώ στην άλλη λαμβάνεται υπόψη η υπάρχουσα ενέργεια του κόμβου και αν ο κόμβος κοντεύει να εξαντλήσει την ενέργειά του δεν αυξάνει την εμβέλεια της ακτίνας του.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Context	Πλαίσιο
Context Awareness	Επίγνωση Πληροφορίας Πλαισίου
Context-Aware Application	Εφαρμογή επίγνωσης πληροφορίας πλαισίου
Context-Aware Systems	Συστήματα επίγνωσης πληροφορίας πλαισίου
Ubiquitous Computing	Πανταχού παρούσα υπολογιστική
Pervasive Computing	Διάχυτη υπολογιστική
Context-aware middleware	Ενδιάμεσο λογισμικό με επίγνωση πλαισίου
Context-aware framework	Βιβλιοθήκη για εφαρμογή επίγνωσης πληροφορίας πλαισίου
Swarm Intelligence	Νοημοσύνη Σμήνους
Particle Swarm Optimization	Βελτιστοποίηση Σμήνους Σωματιδίων
Information's propagation	Διάδοση πληροφορίας
System's Average Radius	Μέση ακτίνα συστήματος
System's Energy Loss	Απώλεια ενέργειας συστήματος
Location awareness	Επίγνωση τοποθεσίας
Activity recognition	Αναγνώριση δραστηριότητας
Particle	Σωματίδιο/Κόμβος
Transmitter	Πομπός
Scalable	Κλιμακούμενος
Direct Sensor Access	Μέθοδος άμεσης πρόσβασης σε αισθητήρα
Middleware infrastructure	Υποδομή ενδιάμεσου λογισμικού
Context Server	Εξυπηρετητής πληροφορίας πλαισίου
Driver	Οδηγός εγκατάστασης προγράμματος
Widget	Είδος λογισμικού
Networked Services	Υπηρεσίες δικτύου
Blackboard model	Μοντέλο διαχείρισης πληροφορίας πλαισίου με διαμοιραζόμενου μέσου
Simulated Annealing	Προσομοιωμένη βαθμιδωτή προσέγγιση
Inertia weight	Αδράνεια βάρους
Fuzzy controller	Ελεγκτής ασάφειας
Cognitive parameter	Γνωσιακή παράμετρος
Social parameter	Κοινωνική παράμετρος
Hardware Sensors	Αισθητήρες υλικού
Hardware Sensing	Ανακάλυψη υλικού
Raw context data	Ακατέργαστα δεδομένα πλαισίου
Aggregation	Συσσωμάτωση
Composition	Σύνθεση
Training phase	Φάση εκπαίδευσης
Subscription	Εγγραφή
Key-Value model	Μοντέλο ζευγαριών κλειδιού τιμής
Matching Algorithm	Αλγόριθμος αντιστοίχισης
Markup scheme model	Μοντέλο μαρκαρίσματος
Graphic Model	Γραφικό μοντέλο

Interface	Διεπαφή
Ontology-based model	Μοντέλο οντολογιών
Context atom	Μονάδα πληροφορίας πλαισίου
Context type	Τύπος πληροφορίας πλαισίου
Context value	Τιμή πληροφορίας πλαισίου
Fitness function	Συνάρτηση καταλληλότητας
Gbest swarm	Σμήνος προτίμησης καθολικά καλύτερου σωματιδίου
Lbest swarm	Σμήνος προτίμησης τοπικά καλύτερου σωματιδίου

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

E.K.Π.A.	Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών
A.M.	Αριθμός Μητρώου
PDA	Personal Digital Assistant
SI	Swarm Intelligence
PSO	Particle Swarm Optimization
W3C	World Wide Web Consortium
GUI	Graphical User Interface
IR	InfraRed
UV	UltraViolet
GPS	Global Positioning System
GSM	Global System for Mobile
RFID	Radio Frequency IDentificator
CC/PP	Composite Capabilities/Preference Profile
UAProf	User Agent Profile
RDF/S	Resource Description Framework Scheme
UML	Unified Modelling Language
CoBRa	Common Broker Architecture
EEXI	Ένωση Ελλήνων Χρηστών Internet
W3C	World Wide Web Consortium
OWL	Web Ontology Language

ΠΑΡΑΡΤΗΜΑ Ι

Παρακάτω παρατίθεται ο πηγαίος κώδικας της υλοποίησης. Αρχικά παρουσιάζεται ο βασικός κορμός της υλοποίησης, ο οποίος είναι ουσιαστικά το σενάριο 1, όπως αυτό παρουσιάστηκε στη παράγραφο 4.2.1, για τη περίπτωση που ακολουθείται ενιαία πολιτική ως προς την αυξομείωση της ακτίνας της κεραίας του κάθε particle. Ακολούθως παρατίθενται οι αλλαγές που πραγματοποιήθηκαν στο βασικό κορμό, με βάση το κάθε σενάριο.

Κώδικας προγράμματος

*/*Αρχείο swarm.c*/*

```
#include <stdio.h>
#include <stdlib.h>
#include "swarm.h"
void Swarm_Init()
{
    int x=0,y=0,num=0,i;
    double temp;

    srand(time(NULL));

    for(x=0;x<SIZE;x++)
    {
        for(y=0;y<SIZE;y++)
        {
            //Initialize radius
            swarm[x][y].radius=1.0;

            //Initialize xcoord
            temp=rand();
            while(temp>=1.0)
                temp/=10;
            swarm[x][y].xcoord=x+temp;

            //Initialize ycoord
            temp=rand();
            while(temp>=1.0)
                temp/=10;
            swarm[x][y].ycoord=y+temp;

            //State defines if the particle has acquired the context
            swarm[x][y].state=FALSE;
            //Each node is given an initial amount of energy
            //such that it cannot extend its radius in an excessive way
            //and try to obtain context from a neighbor
            swarm[x][y].energy=SIZE*0.6;
            //Best radius theoretically should be initially infinitum
            // Practically it is assigned a value that it can never reach
            swarm[x][y].best_radius=swarm[x][y].energy+2;
            swarm[x][y].neighbors=0;
```



```

        swarm[x][y].neighbor=NULL;
        swarm[x][y].actual_time=-1;
        swarm[x][y].quality=0.0;
    }
}

//After determining all particles' coordinates
//each particle neighboring list should be updated
for(x=0;x<SIZE;x++)
    for(y=0;y<SIZE;y++)
        UpdateNeighbors(x,y);

//Determine number of nodes that initially have the context that
//should be desiminated
num=(int)SIZE*SIZE*0.02;
if(num<1)
    num=1;
for(i=1;i<=num;i++)
{
    do{
        x=rand()%SIZE;
        y=rand()%SIZE;
    }while(swarm[x][y].state==TRUE);
    swarm[x][y].state=TRUE;
    swarm[x][y].actual_time=0;
    swarm[x][y].quality=5.0;           //5 levels of quality
}
}

void PSO_Algorithm()
{
    int x=0,y=0,t=0;
    double random_num;
    double new_quality;
    double radius_increasement;
    int time_limit;
    srand(time(NULL));
    char c;

    time_limit=100;
    printf("Time limit=%d\t# nodes=%d\n",time_limit,SIZE*SIZE);
    printf("#nodes with info\t\system's energy\taverage best radius\n");
    for(t=1;t<=time_limit;t++)
    {
        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
            {
                //Periptwsi receiver pou den exei ti pliroforia
                //Case particle does not have the context
                if(swarm[x][y].state==FALSE){
                    random_num=rand();
                    while(random_num>1.0)

```

```

        random_num/=10;
        radius_increasement=max(0,random_num);
        swarm[x][y].radius+=radius_increasement;
        swarm[x][y].energy-=radius_increasement;
    }
    //Check if context was found on the neighborhood
    if(fitness(x,y)==1)
        swarm[x][y].state=TRUE;
    }
    Stats(t);
}
}

int fitness(int x, int y)
{
    UpdateNeighbors(x,y);
    neighborptr temp=swarm[x][y].neighbor;
    if(swarm[x][y].neighbors!=0){
        do{
            if(temp->state==TRUE)
                return 1;
            temp=temp->next;
        }while(temp!=NULL);
    }
    return 0;
}

void UpdateNeighbors(int x, int y)
{
    int i=0,j=0;
    double distance=0.0;
    neighborptr previous=NULL;
    neighborptr temp=NULL;
    for(i=0;i<SIZE;i++)
        for(j=0;j<SIZE;j++)
        {
            distance=(swarm[x][y].xcoord-swarm[i][j].xcoord)*(swarm[x][y].xcoord-
swarm[i][j].xcoord)+(swarm[x][y].ycoord-swarm[i][j].ycoord)*(swarm[x][y].ycoord-
swarm[i][j].ycoord);
            if(distance<0.0)
                distance=-distance;
            if(!areNeighbors(i,j,x,y,previous)){

                if(distance<=swarm[x][y].radius){
                    //Particle [i][j] becomes neighbor with [x][y] and
                    //is added to its neighbors
                    temp=malloc(sizeof(neighborlist));
                    temp->xposition=i;
                    temp->yposition=j;
                    temp->state=swarm[i][j].state;
                    temp->quality=swarm[i][j].quality;
                    temp->next=swarm[x][y].neighbor;
                }
            }
        }
}

```

```

        swarm[x][y].neighbor=temp;
        swarm[x][y].neighbors++;
    }
}
else{
    if(distance>swarm[x][y].radius){
        //Particle [i][j] is no longer neighbor to [x][y] and must be
        //deleted from its neighbors list
        if(previous==NULL){
            temp=swarm[x][y].neighbor;
            swarm[x][y].neighbor=swarm[x][y].neighbor->next;
            free(temp);
        }
        else{
            temp=previous->next;
            previous->next=temp->next;
            free(temp);
        }
        swarm[x][y].neighbors--;
    }
}
}

boolean areNeighbors(int neighbor_x,int neighbor_y,int x,int y,neighborptr
previous)
{
    int i=0;
    neighborptr current;
    current=swarm[x][y].neighbor;
    if(swarm[x][y].neighbors>0){
        do{
            if(current->xposition==neighbor_x      &&      current-
>yposition==neighbor_y)
                return TRUE;
            previous=current;
            current=current->next;
        }while(current!=NULL);
    }
    return FALSE;
}

void Stats(int time)
{
    int x,y;
    double best_radius=0.0;
    double average_energy=0.0;
    int has_info=0;
    for(x=0;x<SIZE;x++)
        for(y=0;y<SIZE;y++)
            {
                if(swarm[x][y].state==TRUE)

```

```

        has_info++;
        best_radius+=swarm[x][y].best_radius;
        average_energy+=swarm[x][y].energy;
    }

    printf("%d\\t\\t\\t\\t%f\\t\\t%f\\n",has_info,(average_energy*100)/(SIZE*SIZE*SIZE*0.6),
best_radius/(SIZE*SIZE));
}

void FinalStatistics(void)
{
    int counter=0;
    int x,y;
    double sum_energy=0.0;
    double average_quality=0.0;
    double best_quality=0.0;
    double worst_quality=0.0;

    printf("\\nTABLE\\n\\n");
    for(x=0;x<SIZE;x++)
    {
        for(y=0;y<SIZE;y++)
        {
            if(swarm[x][y].actual_time==0.0)
                printf("T\\t");
            else
                printf("R\\t");
        }
        printf("\\n");
    }
    printf("\\nQUALITY TABLE\\n\\n");
    for(x=0;x<SIZE;x++)
    {
        for(y=0;y<SIZE;y++)
        {
            if(swarm[x][y].energy<0)
                counter++;
            sum_energy+=swarm[x][y].energy;
            printf("%.3f\\t",swarm[x][y].quality);
            average_quality+=swarm[x][y].quality;
            if(swarm[x][y].quality>best_quality)
                best_quality=swarm[x][y].quality;
            if(swarm[x][y].quality<worst_quality)
                worst_quality=swarm[x][y].quality;
        }
        printf("\\n");
    }
    printf("\\nPOWER TABLE\\n\\n");
    for(x=0;x<SIZE;x++)
    {
        for(y=0;y<SIZE;y++)
            printf("%.3f\\t",swarm[x][y].energy);
    }
}

```

```

        printf("\n");
    }
    printf("\n");
    printf("%d nodes lost all their power\n",counter);
    printf("System's initial power=%.3f\n",SIZE*SIZE*SIZE*0.6);
    printf("System's remaining power=%.3f\n",sum_energy);
    printf("System's average quality=%.3f\n",average_quality/(SIZE*SIZE));
    printf("Particles' worst quality=%.3f\n",worst_quality);
    printf("Particles' best quality=%.3f\n",best_quality);
}

void ReleaseMemory()
{
    int i=0,j=0;
    neighborptr temp=NULL;
    for(i=0;i<SIZE;i++)
        for(j=0;j<SIZE;j++)
            while(swarm[i][j].neighbor!=NULL)
                {
                    temp=swarm[i][j].neighbor;
                    swarm[i][j].neighbor=swarm[i][j].neighbor->next;
                    free(temp);
                    temp=NULL;
                }
}

double max(double x, double y)
{
    if(x>=y)
        return x;
    return y;
}

/*Αρχείο swarm.h*/

#ifndef __SWARM__
#define __SWARM__

#define SIZE 10
#define OK      100
#define MINIMUM_ENERGY SIZE*0.12 //Minimum energy is defined as 20% of initial
power
int errNo;
double best_radius;
typedef enum {FALSE,TRUE} boolean;

typedef struct neighborlist* neighborptr;

typedef struct neighborlist{
    int xposition;
    int yposition;
    boolean state;
}

```

```
    double quality;
    neighborptr next;
}neighborlist;

typedef struct particle{
    double xcoord; //x coordinate
    double ycoord; //y coordinate
    double radius;
    double energy; //remaining energy
    int actual_time; //The time context was acquired
    double best_radius; //local best
    boolean state;
    double quality;
    neighborptr neighbor;
    int neighbors;
}particle;

particle swarm [SIZE][SIZE];

void PrintError();
void Swarm_Init();
void Stats(int time);
void PSO_Algorithm();
int fitness(int x, int y);
void UpdateNeighbors(int x, int y);
boolean areNeighbors(int neighbor_x,int neighbor_y,int x, int y,neighborptr
previous);
void FinalStatistics();
void ReleaseMemory();
double max(double x, double y);

#endif

/*main.c*/

#include <stdio.h>
#include <stdlib.h>
#include "swarm.h"

int main()
{
    Swarm_Init();
    PSO_Algorithm();
    FinalStatistics();
    ReleaseMemory();
}
```

Απαιτούμενες αλλαγές για το 1^ο σενάριο, όταν δεν ακολουθείται ενιαία πολιτική ως προς την αυξομείωση της εμβέλειας των κεραιών.

```

void PSO_Algorithm()
{
    int x=0,y=0,t=0;
    double random_num;
    double new_quality;
    double radius_increasement;
    int time_limit;
    srand(time(NULL));
    char c;

    time_limit=100;
    printf("Time limit=%d\t# nodes=%d\n",time_limit,SIZE*SIZE);
    printf("#nodes with info\tssystem's energy\taverage best radius\n");
    for(t=1;t<=time_limit;t++)
    {
        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
            {
                //Case particle does not have the context
                if(swarm[x][y].state==FALSE){
                    random_num=rand();
                    while(random_num>1.0)
                        random_num/=10;
                    radius_increasement=max(0,random_num);
                    //1st policy
                    if(swarm[x][y].energy>=MINIMUM_ENERGY){
                        swarm[x][y].radius+=radius_increasement;
                        swarm[x][y].energy-=radius_increasement;
                    }
                    //2nd policy
                    else if(swarm[x][y].energy>0.5 &&
swarm[x][y].energy>=radius_increasement/2){
                        swarm[x][y].radius+=radius_increasement/2;
                        swarm[x][y].energy-=radius_increasement/2;
                    }
                    //Check if context was found on the neighborhood
                    if(fitness(x,y)==1)
                        swarm[x][y].state=TRUE;
                }
            }
        Stats(t);
    }
}

```

Απαιτούμενες αλλαγές για το 2^ο σενάριο

```

void PSO_Algorithm()
{
    int x=0,y=0,t=0;
    double random_num;
    double new_quality;
    double radius_increasement;
    int time_limit;
    srand(time(NULL));
    int period;
    char c;

    time_limit=100;
    printf("Time limit=%d\t# nodes=%d\n",time_limit,SIZE*SIZE);
    printf("#nodes with info\tssystem's energy\taverage best radius\n");

    period=rand()%7;
    while(period==0 || period==1)
        period=rand()%7;

    for(t=1;t<=time_limit;t++)
    {
        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
            {
                //Case particle is one the initial transmitters
                if(swarm[x][y].actual_time==0){
                    swarm[x][y].quality=(-5.0/period)*(t%period)+5.0;
                    //Transmitters re-emit context to the swarm
                    if(swarm[x][y].quality<=0.0)
                        swarm[x][y].quality=5.0;
                }
                else{
                    new_quality=(double)((-
5.0/period)*(t%period)+swarm[x][y].quality);
                    swarm[x][y].quality=new_quality;
                    if(swarm[x][y].quality<=0.0)
                        swarm[x][y].quality=0.0;
                }
            }

        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
            {
                random_num=(double)rand();
                while(random_num>1.0)
                    random_num/=10;
                radius_increasement=max(0,random_num);

                //Case context is not yet acquired by the particle
                if(swarm[x][y].state==FALSE){

```



```

        if( swarm[x][y].energy>0.0){
            if(swarm[x][y].energy>radius_increasement){
                swarm[x][y].radius+=radius_increasement;
                swarm[x][y].energy-=radius_increasement;
            }
            else{
                swarm[x][y].radius+=swarm[x][y].energy;
                swarm[x][y].energy=0.0;
            }
        }

        if(fitness(x,y)==1){
            swarm[x][y].state=TRUE;
            if(swarm[x][y].radius<swarm[x][y].best_radius)
                swarm[x][y].best_radius=swarm[x][y].radius;
            swarm[x][y].actual_time=t;
        }
    }
    else{
        //Case context is acquired by the particle
        if(swarm[x][y].actual_time!=0){

            if(swarm[x][y].energy>MINIMUM_ENERGY                &&
            swarm[x][y].radius>1.0){

                swarm[x][y].radius-=radius_increasement;

                if(fitness(x,y)==1){

                    if(swarm[x][y].radius<swarm[x][y].best_radius)

                        swarm[x][y].best_radius=swarm[x][y].radius;
                        swarm[x][y].actual_time=t;
                    }
                    else{
                        swarm[x][y].state=FALSE;
                        swarm[x][y].quality=0.0;
                    }
                }
            }
        }
    }
}
Stats(t);
}
}

int fitness(int x, int y)
{
    boolean met_first=FALSE;
    int return_value=0;
    UpdateNeighbors(x,y);
    neighborptr temp=swarm[x][y].neighbor;

```

```

    if(swarm[x][y].neighbors!=0){
        do{
            //If a neighbor was found
            if(temp->state==TRUE){
                //if it is the first neighbor
                if(!met_first){
                    met_first=TRUE;
                    swarm[x][y].quality=temp->quality;
                }
                //Direct each particle to it's neighbor with the best quality context
                else{
                    if(swarm[x][y].quality<temp->quality)
                        swarm[x][y].quality=temp->quality;
                }
                return_value=1;
            }
            temp=temp->next;
        }while(temp!=NULL);
    }
    return return_value;
}

```

Απαιτούμενες αλλαγές για το 2^ο σενάριο, όταν δεν ακολουθείται ενιαία πολιτική ως προς την αυξομείωση της εμβέλειας των κεραιών.

```

void PSO_Algorithm()
{
    int x=0,y=0,t=0;
    double random_num;
    double new_quality;
    double radius_increasement;
    int time_limit;
    srand(time(NULL));
    int period;
    char c;

    time_limit=100;
    printf("Time limit=%d\t# nodes=%d\n",time_limit,SIZE*SIZE);
    printf("#nodes with info\t\system's energy\taverage best radius\n");

    period=rand()%7;
    while(period<3)
        period=rand()%7;

    for(t=1;t<=time_limit;t++)
    {
        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
            {
                //Case particle is one the initial transmitters
                if(swarm[x][y].actual_time==0){

```

```

        swarm[x][y].quality=((double)(-
5.0/period))*(t%period)+5.0;
        //Transmitters re-emit context to the swarm
        if(swarm[x][y].quality<=0.0)
            swarm[x][y].quality=5.0;
    }
    else{
        new_quality=(double)((-
5.0/period)*(t%period)+swarm[x][y].quality);
        swarm[x][y].quality=new_quality;
        if(swarm[x][y].quality<=0.0)
            swarm[x][y].quality=0.0;
    }
}

for(x=0;x<SIZE;x++)
    for(y=0;y<SIZE;y++)
    {
        random_num=(double)rand();
        while(random_num>1.0)
            random_num/=10.0;

        radius_increasement=max(0,random_num);

        //Case context is not yet acquired by the particle
        if(swarm[x][y].state==FALSE){

            //1st policy
            if(swarm[x][y].energy>=MINIMUM_ENERGY){
                swarm[x][y].radius+=radius_increasement;
                swarm[x][y].energy-=radius_increasement;
            }
            //2nd policy
            else if(swarm[x][y].energy>0.5 &&
swarm[x][y].energy>=radius_increasement/2){
                swarm[x][y].radius+=radius_increasement/2;
                swarm[x][y].energy-=radius_increasement/2;
            }
            if(fitness(x,y)==1){
                swarm[x][y].state=TRUE;
                if(swarm[x][y].radius<swarm[x][y].best_radius)
                    swarm[x][y].best_radius=swarm[x][y].radius;
                swarm[x][y].actual_time=t;
            }
        }
        else{
            //Case context is acquired by the particle
            if(swarm[x][y].actual_time!=0){
                if(swarm[x][y].energy>MINIMUM_ENERGY &&
swarm[x][y].radius>1.0 && swarm[x][y].quality<=0.0){
                    swarm[x][y].radius-=radius_increasement;

```

```

        if(fitness(x,y)==1){
            if(swarm[x][y].radius<swarm[x][y].best_radius)
                swarm[x][y].best_radius=swarm[x][y].radius;
                swarm[x][y].actual_time=t;
            }
            else{
                swarm[x][y].state=FALSE;
                swarm[x][y].quality=0.0;
            }
        }
    }
}
Stats(t);
}
}

```

Απαιτούμενες αλλαγές για το 3^ο σενάριο, όπου δεν ακολουθείται ενιαία πολιτική ως προς την αυξομείωση της εμβέλειας των κεραιών.

Σε αυτό το σενάριο έγιναν αλλαγές και στο αρχείο κεφαλίδας και στο πηγαίο κώδικα

```

/*swarm.h*/
typedef struct particle{
    double xcoord; //x coordinate
    double ycoord; //y coordinate
    int neighborcoords[2];
    double radius;
    double energy; //remaining energy
    int actual_time; //The time we got the info
    double best_radius; //local best
    boolean state;
    double quality;
    neighborptr neighbor;
    int neighbors;
}particle;

/*swarm(with time limit).c*/

void Swarm_Init()
{
    int x=0,y=0,num=0,i;
    double temp;
    srand(time(NULL));

    for(x=0;x<SIZE;x++)
    {
        for(y=0;y<SIZE;y++)
        {
            //Initialize radius

```

```

        swarm[x][y].radius=1;
        //Initialize xcoord
        temp=rand();
        while(temp>=1.0)
            temp/=10;
        swarm[x][y].xcoord=x+temp;
        //Initialize ycoord
        temp=rand();
        while(temp>=1.0)
            temp/=10;
        swarm[x][y].ycoord=y+temp;
        //Initialize other variables
        swarm[x][y].state=FALSE;
        swarm[x][y].energy=SIZE*0.6;
        swarm[x][y].best_radius=swarm[x][y].energy+2;
        swarm[x][y].neighbors=0;
        swarm[x][y].neighbor=NULL;
        swarm[x][y].actual_time=-1;
        swarm[x][y].quality=0.0;
        swarm[x][y].neighborcoords[0]=-1;
        swarm[x][y].neighborcoords[1]=-1;
    }
}

for(x=0;x<SIZE;x++)
    for(y=0;y<SIZE;y++)
        UpdateNeighbors(x,y);

num=(int)SIZE*SIZE*0.02;
if(num<1)
    num=1;
for(i=1;i<=num;i++)
{
    do{
        x=rand()%SIZE;
        y=rand()%SIZE;
    }while(swarm[x][y].state==TRUE);
    swarm[x][y].state=TRUE;
    swarm[x][y].actual_time=0;
    swarm[x][y].quality=5.0;
}
errNo=0;
}

void PSO_Algorithm()
{
    int x=0,y=0,t=0;
    double random_num;
    double new_quality;
    double radius_increasement;
    int time_limit,delay_time,period;
    int check=0;

```

```

srand(time(NULL));

time_limit=100;
printf("Time limit=%d\t# nodes=%d\n",time_limit,SIZE*SIZE);
printf("#nodes with info\tssystem's energy\taverage best radius\n");

period=rand()%7;
while(period==0 || period==1)
    period=rand()%7;

delay_time=(int) 0.3*time_limit;
for(t=1;t<=time_limit;t++)
{
    for(x=0;x<SIZE;x++)
        for(y=0;y<SIZE;y++)
            {
                //Case particle is one the initial transmitters
                if(swarm[x][y].actual_time==0){
                    swarm[x][y].quality=(-5.0/period)*(t%period)+5.0;
                    //Transmitters re-emit context to the swarm
                    if(swarm[x][y].quality<=0.0)
                        swarm[x][y].quality=5.0;
                }
                else{
                    new_quality=(double)((-5.0/period)*t+swarm[x][y].quality);
                    swarm[x][y].quality=new_quality;
                    if(swarm[x][y].quality<=0.0){
                        swarm[x][y].quality=0.0;
                        swarm[x][y].state=FALSE;
                    }
                }
            }
        for(x=0;x<SIZE;x++)
            for(y=0;y<SIZE;y++)
                {
                    //Case context is not yet acquired by the particle
                    if(swarm[x][y].state==FALSE){
                        random_num=rand();
                        while(random_num>1.0)
                            random_num/=10;

                        radius_increasement=max(0,random_num);
                        //1st policy
                        if(swarm[x][y].energy>=MINIMUM_ENERGY){
                            swarm[x][y].radius+=radius_increasement;
                            swarm[x][y].energy-=radius_increasement;
                        }
                        //2nd policy
                        else if(swarm[x][y].energy>0.5 &&
swarm[x][y].energy>=radius_increasement/2){
                            swarm[x][y].radius+=radius_increasement/2;
                            swarm[x][y].energy-=radius_increasement/2;

```

```

    }
    if((check=fitness(x,y))!=0){
        swarm[x][y].state=TRUE;
        if(swarm[x][y].radius<swarm[x][y].best_radius)
            swarm[x][y].best_radius=swarm[x][y].radius;
        swarm[x][y].actual_time=t;
    }
}
else{
    //Case context is acquired by the particle
    if(swarm[x][y].actual_time!=0){
        if((t-swarm[x][y].actual_time)>delay_time           ||
swarm[x][y].quality<=0.0){
            swarm[x][y].state=FALSE;
            swarm[x][y].actual_time=-1;
            swarm[x][y].quality=0.0;
            swarm[x][y].neighborcoords[0]=-1;
            swarm[x][y].neighborcoords[1]=-1;
        }
    }
}
}
}
Stats(t);
}
}

int fitness(int x, int y)
{
    int return_value=0;
    UpdateNeighbors(x,y);
    neighborptr temp=swarm[x][y].neighbor;
    if(swarm[x][y].neighbors!=0){
        do{
            //Otan vrw enan geitona
            if(temp->state==TRUE){
                if(temp->xposition!=swarm[x][y].neighborcoords[0] || temp-
>yposition!=swarm[x][y].neighborcoords[1]){
                    if(swarm[x][y].quality<temp->quality)
                    {
                        swarm[x][y].quality=temp->quality;
                        swarm[x][y].neighborcoords[0]=temp->xposition;
                        swarm[x][y].neighborcoords[1]=temp->yposition;
                    }
                    return_value=1;
                }
            }
            temp=temp->next;
        }while(temp!=NULL);
    }
    return return_value;
}

```

ΑΝΑΦΟΡΕΣ

- [1] G.D. Abowd et al., "Towards a better understanding of context and context-awareness", *Handheld and Ubiquitous Computing, LNCS 1707*, Springer, 1999, pp. 304-307.
- [2] M. Baldauf, S. Dustdar, F. Rosenberg, "A survey of context aware systems", *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, 2007, pp.263-277.
- [3] A.K. Dey, "Understanding and Using Context", *Personal and Ubiquitous Computing*, vol. 5, issue 1, Springer-Verlag, February 2001 pp 4 – 7
- [4] N. Nedjah, L. de M. Mourelle, *Swarm Intelligent Systems, Studies in Computational Intelligence*, vol. 26, Springer-Verlag, 2006
- [5] A. Abraham, C. Grosan, V.Ramos, *Swarm Intelligence in Data Mining, Studies in Computational Intelligence*, vol. 34, Springer-Verlag, 2006
- [6] http://en.wikipedia.org/wiki/Context_awareness [Προσπελάστηκε 15/03/2009]
- [7] B. Schilit, M. Theimer, "Disseminating Active Map Information to Mobile Hosts", *IEEE Network*, vol. 8, no. 5, Sept. 1994, pp. 22-32
- [8] P.J. Brown, J.D. Bovey, X. Chen, "Context-Aware Applications: From the Laboratory to the Marketplace", *IEEE Personal Communications*, vol. 4, no. 5, 1997, pp. 58-64
- [9] N. Ryan, J. Pascoe, D. Morse, "Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant", Gaffney, V. et al., eds., *Computer Applications in Archaeology*, 1997
- [10] B. Schilit, N. Adams, R. Want, "Context-Aware Computing Applications", *1st Int'l Workshop on Mobile Comp. Syst. and App. (WMCSA 1994)*, pp. 85-90
- [11] A.K. Dey, G.D. Abowd, A. Wood, "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services" *Knowledge-Based Systems*, vol. 11, Sept. 1998, pp. 3-13
- [12] J. Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers", *2nd Int'l Symp. on Wearable Computers (ISWC 1998)*, pp. 92-99
- [13] P.J. Brown, "The Stick-e Document: a Framework for Creating Context-Aware Applications", *Proc. Electronic Publishing '96*, 1996, pp. 259-272
- [14] D. Franklin, J. Flaschbart, "All Gadget and No Representation Makes Jack a Dull Environment", *Proc. AAAI 1998 Spring Symposium on Intelligent Environments*, AAAI TR SS-98-02, 1998, pp. 155-160
- [15] A. Ward, A. Jones, A. Hopper, "A New Location Technique for the Active Office", *IEEE Personal Communications*, vol. 4, no. 5, 1997, pp. 42-47
- [16] T. Rodden, K. Cheverst, K. Davies, A. Dix, "Exploiting Context in HCI Design for Mobile Systems", *1st Workshop on Human Computer Interaction with Mobile Devices (1998)*
- [17] R. Hull, P. Neaves, J. Bedford-Roberts, "Towards Situated Computing", *1st Int'l Symp. on Wearable Computers (ISWC 1997)*, pp. 146-153
- [18] A. Schmidt, M. Beigl, H.W. Gellersen, "There is more to Context than Location" *Computers & Graphics Journal*, Elsevier vol. 23, no. 6, Dec. 1999, pp. 893-902
- [19] A. Schmidt, "Ubiquitous Computing - Computing in Context". doctoral dissertation, Univ. Lancaster, 2002
- [20] J. Kennedy, R. C. Eberhart, Y. Shi., "Swarm Intelligence", Morgan Kaufmann, San Francisco, CA, 2001