

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Μελέτη και ανάπτυξη αλγορίθμων εμπνευσμένων
από τη Βιολογία (Bio-inspired algorithms)»**

ΔΗΜΗΤΡΙΟΣ Ν. ΚΑΡΒΟΥΝΑΣ

Κατεύθυνση: Ψηφιακές Επικοινωνίες και Δίκτυα

Επιβλέπων: **Π. Δεμέστιχας**

Αναπληρωτής Καθηγητής

Πειραιάς, 2009

РАНЕЕЗНАМО ПЕРПАА

РАНЕЕЗНАМО ПЕРПАА

РАНЕЕЗНАМО ПЕРПАА

Περίληψη

Τα έντομα τα οποία ζουν σε αποικίες, και ιδίως τα μυρμήγκια, έχουν κεντρίσει το ενδιαφέρον των επιστημόνων εδώ και πολλά χρόνια λόγω της εκπληκτικής συλλογικής συμπεριφοράς που παρουσιάζουν. Μάλιστα έχουν εμπνεύσει αρκετές μεθόδους και τεχνικές, με πιο γνωστή και ευρέως διαδεδομένη την τεχνική βελτιστοποίησης "Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization - ACO)". Η ACO εμπνέεται από τον τρόπο που τα μυρμήγκια συλλέγουν την τροφή τους. Μόλις ένα μυρμήγκι εντοπίσει την τροφή, εκκρίνει στο έδαφος μια χημική ουσία, την φερομόνη. Τα υπόλοιπα μυρμήγκια έχουν την ικανότητα να εντοπίσουν την φερομόνη που άφησε το προηγούμενο μυρμήγκι κι έτσι να ακολουθήσουν τη διαδρομή μέχρι την τροφή. Αν υπάρχουν περισσότερες από μία διαδρομές προς την τροφή, τότε στην πιο σύντομη θα εναποτίθεται περισσότερη ποσότητα φερομόνης διότι τα μυρμήγκια θα επιστρέφουν πιο γρήγορα. Έτσι τα περισσότερα μυρμήγκια ακολουθούν την διαδρομή με την περισσότερη ποσότητα φερομόνης κι επομένως την πιο σύντομη.

Μία επίσης πολύ γνωστή μέθοδος βελτιστοποίησης είναι η Προσομοιωμένη Ανόπτηση (Simulated Annealing). Η διαδικασία ανόπτησης αποτελεί τη φυσική διεργασία της έντονης θέρμανσης ενός στερεού και κατόπιν της ψύξης του μέχρι αυτό να κρυσταλλώσει. Τα άτομα μέσα στο υλικό έχουν υψηλές ενέργειες σε υψηλές θερμοκρασίες και επομένως έχουν περισσότερη ελευθερία. Ενώ η θερμοκρασία μειώνεται, μειώνονται παράλληλα και οι ατομικές ενέργειες. Ένας κρύσταλλος με ομαλή δομή επιτυγχάνεται στην κατάσταση όπου το σύστημα έχει ελάχιστη ενέργεια. Εάν η ψύξη γίνει πολύ γρήγορα, εκτεταμένες ανωμαλίες παρατηρούνται στην κρυσταλλική δομή. Το σύστημα δεν φτάνει στην κατάσταση ελάχιστης ενέργειας και καταλήγει σε μια πολυκρυσταλλική δομή η οποία έχει υψηλότερη ενέργεια από την ελάχιστη.

Στα πλαίσια της εργασίας που παρουσιάζεται, έγινε ένας συνδυασμός αυτών των δύο τεχνικών βελτιστοποίησης και δημιουργήθηκε ένας υβριδικός αλγόριθμος Simulated Annealing με στοιχεία Ant Colony Optimization. Ο λόγος της επιλογής των δύο αλγορίθμων είναι ότι είναι ευρέως γνωστοί και δοκιμασμένοι στον κόσμο των τηλεπικοινωνιών, και χρησιμοποιούνται ακόμα και για την επίλυση των πιο σύγχρονων προβλημάτων.

Για τη μελέτη του αλγορίθμου χρησιμοποιήθηκε το πρόβλημα των Ρομποτικών, Γνωσιακών Σημείων Πρόσβασης (Robotic, Cognitive Access Points - RoboCAPs). Τα RoboCAPs είναι Access Points (APs) που έχουν τη δυνατότητα να κινούνται αυτόματα και αυτόνομα (μέσω ρομποτικών πλατφόρμων ή τοποθετημένα σε αυτοκίνητα), να καθορίζουν τις θέσεις τους και να δημιουργούν ad-hoc δίκτυα για την παροχή υπηρεσιών στους χρήστες. Σκοπός του προβλήματος είναι η εύρεση της

κατανομής των RoboCAPs στις κορυφές ενός γράφου, έτσι ώστε να ελαχιστοποιείται η αντικειμενική συνάρτηση η οποία συνδέεται με το κόστος μετακίνησης των RoboCAPs και το τηλεπικοινωνιακό κόστος της σύνδεσης μεταξύ των RoboCAPs και μεταξύ των RoboCAPs και των χρηστών. Τέλος, τα αποτελέσματα που εξάγει ο αλγόριθμος συγκρίνονται με αυτά του απλού αλγορίθμου SA και της εξαντλητικής αναζήτησης.

Λέξεις κλειδιά: κοινωνικά έντομα, μυρμήγκια, αποικία μυρμηγκιών, βελτιστοποίηση, ant colony, optimization, ad-hoc, γνωσιακά δίκτυα, προσομοιωμένη απόπτωση, simulated annealing, ρομποτικά, γνωσιακά σημεία πρόσβασης, robotic, cognitive, access points, υβριδικός, αλγόριθμος.

1 ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

2	ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ	11
3	ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ	15
4	ΚΟΙΝΩΝΙΚΑ ΕΝΤΟΜΑ.....	19
4.1	Εισαγωγή	19
4.2	Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO)	23
4.2.1	Έμπνευση από τη βιολογία	24
4.2.2	Η τεχνική της βελτιστοποίησης.....	26
4.2.3	ACO για το Πρόβλημα του Πλανόδιου Πωλητή (Traveling Salesman Problem - TSP)	26
4.2.4	Η Μεταε启发τική Βελτιστοποίηση Αποικίας Μυρμηγκιών (The Ant Colony Optimization Metaheuristic)	27
4.2.4.1	Ένα συνδυαστικό πρόβλημα βελτιστοποίησης	28
4.2.5	Κύριοι ACO αλγόριθμοι.....	30
4.2.5.1	Σύστημα Μυρμηγκιών (Ant System – AS).....	30
4.2.5.2	MAX-MIN Σύστημα Μυρμηγκιών (MAX-MIN Ant System – MMAS)	31
4.2.5.3	Σύστημα Αποικίας Μυρμηγκιών (Ant Colony System – ACS)	32
5	ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ (SIMULATED ANNEALING).....	35
5.1	Εισαγωγή	35
5.2	Βασικά στοιχεία	35
6	AD-HOC ΔΙΚΤΥΑ	39
6.1	Ορισμός και ιδιότητες των Ad-hoc Δικτύων	39
6.2	Λειτουργία των Ad-hoc δικτύων	39
6.3	Εφαρμογές των Ad-hoc Δικτύων.....	40
6.4	Χαρακτηριστικά των Ad-hoc δικτύων	41
6.5	Ποιότητα υπηρεσιών στα Ad-hoc δίκτυα	42
6.6	Προβλήματα, προκλήσεις	44
6.7	Mobile Ad-hoc Networks	46
6.7.1	Εισαγωγή.....	46
6.7.2	Χαρακτηριστικά των MANETS	46

7	ΓΝΩΣΙΑΚΑ ΔΙΚΤΥΑ	49
7.1	Εισαγωγή	49
7.2	Κίνητρο: Γνωσιακά Δίκτυα και η Διαχειριστική Λειτουργικότητά τους	50
7.3	Περιγραφή του προβλήματος	51
7.3.1	Πλαίσιο	52
7.3.2	Προφίλ	52
7.3.3	Πολιτικές	52
7.3.4	Έξοδος	53
7.3.5	Γνωσιακά Χαρακτηριστικά	53
7.4	Λειτουργικότητα Διαχείρισης για Γνωσιακά Σημεία Πρόσβασης	54
7.4.1	Αυτόνομη Διαχείριση Σημείων Πρόσβασης (Autonomic Management of Access Points – AMAP)	54
7.4.1.1	Εισαγωγή και Λειτουργική Ανάλυση	54
7.4.1.2	Προφίλ και Συμφωνίες	55
7.4.1.3	Παρακολούθηση, ανίχνευση και απόκτηση πλαισίου	55
7.4.1.4	Μεσιτεία Πόρων και Υπηρεσιών	56
7.4.1.5	Διαπραγμάτευση Διαμόρφωσης, Επιλογή και Εκτέλεση	56
7.4.1.6	Γνωσιακά Χαρακτηριστικά	56
8	ΡΟΜΠΟΤΙΚΑ, ΓΝΩΣΙΑΚΑ ΣΗΜΕΙΑ ΠΡΟΣΒΑΣΗΣ (ROBOTIC, COGNITIVE ACCESS POINTS - ROBOCAPS)	59
8.1	Η ιδέα των RoboCAPs	59
8.2	Στόχος των RoboCAPs	61
8.3	Μαθηματικό μοντέλο	67
9	ΠΡΟΣΟΜΟΙΩΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ROBOCAPS	71
9.1	Εισαγωγή στο πρόβλημα των RoboCAPs	71
9.2	Περιγραφή των αλγορίθμων	71
9.2.1	Αλγόριθμος Simulated Annealing	72
9.2.2	Αλγόριθμος Simulated Annealing με στοιχεία Ant Colony Optimization	72
9.3	Οδηγός χρήσης του προγράμματος εξομοίωσης	74
9.4	Παραδείγματα	76
9.4.1	Παράδειγμα 1	77
9.4.1.1	Λύση μέσω εξαντλητικής αναζήτησης	78
9.4.1.2	Λύση με χρήση του αλγορίθμου Simulated Annealing	78
9.4.1.3	Λύση με χρήση του αλγορίθμου Simulated Annealing με στοιχεία ACO	79
9.4.2	Παράδειγμα 2	81
9.4.2.1	Λύση μέσω εξαντλητικής αναζήτησης	81

9.4.2.2	Λύση με χρήση του αλγορίθμου Simulated Annealing	82
9.4.2.3	Λύση με χρήση του αλγορίθμου Simulated Annealing με στοιχεία ACO	83
9.4.3	Παράδειγμα 3.....	85
9.4.3.1	Λύση μέσω εξαντλητικής αναζήτησης.....	85
9.4.3.2	Λύση με χρήση του αλγορίθμου Simulated Annealing	86
9.4.3.3	Λύση με χρήση του αλγορίθμου Simulated Annealing με στοιχεία ACO	86
9.4.4	Παράδειγμα 4.....	88
9.4.4.1	Λύση μέσω του αλγορίθμου Simulated Annealing.....	89
9.4.4.2	Λύση μέσω του αλγορίθμου Simulated Annealing με στοιχεία ACO	90
9.5	Συμπεράσματα – Παρατηρήσεις.....	91
10	ΠΑΡΑΡΤΗΜΑ	93
10.1	Κλάση InitProblem.....	93
10.2	Πηγαίος κώδικας απλού αλγορίθμου SA.....	102
10.3	Πηγαίος κώδικας υβριδικού SA-ACO.....	108
11	ΑΝΑΦΟΡΕΣ - ΒΙΒΛΙΟΓΡΑΦΙΑ	121

РАНЕЕЗНАМО ПЕРПАА

2 ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 4.1 Μυρμήγκια <i>Atta</i> μεταφέρουν φύλλα πίσω στη φωλιά.....	19
Εικόνα 4.2 Αλυσίδες από μυρμήγκια <i>Osecorhylla</i>	20
Εικόνα 4.3 Δύο εργάτες κρατάνε μια κάμπια στη γνάθο.....	20
Εικόνα 4.4 (α) Οι γέφυρες έχουν ίδιο μήκος. (β) Οι γέφυρες έχουν διαφορετικό μήκος.....	26
Εικόνα 4.5 Ένα μυρμήγκι στην πόλη i , αποφασίζει την επόμενη πόλη που θα επισκεφτεί μέσω ενός στοχαστικού μηχανισμού: αν δεν έχει επισκεφτεί προηγουμένως την j , τότε μπορεί να την επιλέξει με μια πιθανότητα που είναι ανάλογη της φερομόνης που έχει αντιστοιχηθεί στην πλευρά (i,j)	27
Εικόνα 4.6 Εικόνα πιθανού γράφου της κατασκευής για ένα TSP με τέσσερις πόλεις όπου οι συνιστώσες είναι συσχετισμένες με τις πλευρές.	29
Εικόνα 5.1 Διάγραμμα ροής ενός standard simulated annealing αλγόριθμου.	37
Εικόνα 6.1 Ασύρματο Ad-hoc δίκτυο.....	40
Εικόνα 6.2 Κινητό ad-hoc δίκτυο.	46
Εικόνα 7.1 Βρόχος ανάδρασης γνωσιακών δικτύων.....	50
Εικόνα 7.2 Λειτουργικότητα διαχείρισης για τμήματα γνωστικού δικτύου.	52
Εικόνα 7.3 Λειτουργική Ανάλυση μιας AMAP οντότητας.	55
Εικόνα 8.1 Αρχική αρχιτεκτονική των RoboCAPs και περιοχές εφαρμογής.	61
Εικόνα 8.2 Στιγμιότυπο της ιδέας της δικτύωσης των RoboCAPs (οι RATs που υποδεικνύονται είναι πιθανές λύσεις).....	62
Εικόνα 8.3 Απόκτηση και ανταλλαγή πληροφοριών πλαισίου, προφίλ και πολιτικών, και σχετικής γνώσης στα RoboCAPs.	64
Εικόνα 8.4 Υψηλού επιπέδου λειτουργική αρχιτεκτονική για από κοινού βελτιστοποίηση του ράδιο-δικτύου και της κίνησης των RoboCAP οντοτήτων, ώστε να γίνει χειρισμός του πλαισίου που αντιμετωπίζεται, βασισμένο επίσης σε ικανότητες (προφίλ) και πολιτικές.	65
Εικόνα 8.5 Παράδειγμα λειτουργίας των RoboCAPs.	66

Εικόνα 9.1 (α) Αρχική οθόνη. (β) Εισαγωγή συντεταγμένων των κόμβων.	75
Εικόνα 9.2 Εισαγωγή αποστάσεων συνδεδεμένων κόμβων.....	75
Εικόνα 9.3 (α) Ρύθμιση των RoboCAPs. (β) Ρύθμιση των χρηστών.	76
Εικόνα 9.4 Γράφος του Προβλήματος 1.	77
Εικόνα 9.5 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 1.....	79
Εικόνα 9.6 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 1.....	80
Εικόνα 9.7 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.	80
Εικόνα 9.8 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.....	80
Εικόνα 9.9 Γράφος του Προβλήματος 2.	81
Εικόνα 9.10 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 2.....	83
Εικόνα 9.11 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 2.....	84
Εικόνα 9.12 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.....	84
Εικόνα 9.13 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.....	84
Εικόνα 9.14 Γράφος του Προβλήματος 3.....	85
Εικόνα 9.15 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 3.....	86
Εικόνα 9.16 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 3.....	87
Εικόνα 9.17 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.....	88
Εικόνα 9.18 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.....	88
Εικόνα 9.19 Ο γράφος για το Πρόβλημα 4.....	89
Εικόνα 9.20 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 4.....	90
Εικόνα 9.21 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 4.....	90

Εικόνα 9.22 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.....91

Εικόνα 9.23 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.....91

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

РАНЕЕЗНАМО ПЕРПАА

3 ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Πίνακας 9.1 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.....	78
Πίνακας 9.2 Λεπτομέρειες χρηστών του Προβλήματος 1.....	78
Πίνακας 9.3 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.	78
Πίνακας 9.4 Συνδεδεμένα RoboCAPs του Προβλήματος 1.....	78
Πίνακας 9.5 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.....	78
Πίνακας 9.6 Λεπτομέρειες χρηστών του Προβλήματος 1.....	78
Πίνακας 9.7 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.	78
Πίνακας 9.8 Συνδεδεμένα RoboCAPs του Προβλήματος 1.....	78
Πίνακας 9.9 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.....	79
Πίνακας 9.10 Λεπτομέρειες χρηστών του Προβλήματος 1.....	79
Πίνακας 9.11 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.	79
Πίνακας 9.12 Συνδεδεμένα RoboCAPs του Προβλήματος 1.....	79
Πίνακας 9.13 Πίνακας φερομόνης του Προβλήματος 1.	79
Πίνακας 9.14 Πίνακας πιθανοτήτων του Προβλήματος 1.	80
Πίνακας 9.15 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.....	81
Πίνακας 9.16 Λεπτομέρειες χρηστών του Προβλήματος 2.....	81
Πίνακας 9.17 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.	82
Πίνακας 9.18 Συνδεδεμένα RoboCAPs του Προβλήματος 2.....	82
Πίνακας 9.19 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.....	82
Πίνακας 9.20 Λεπτομέρειες χρηστών του Προβλήματος 2.....	82
Πίνακας 9.21 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.	82
Πίνακας 9.22 Συνδεδεμένα RoboCAPs του Προβλήματος 2.....	82
Πίνακας 9.23 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.....	83

Πίνακας 9.24 Λεπτομέρειες χρηστών του Προβλήματος 2.....	83
Πίνακας 9.25 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.	83
Πίνακας 9.26 Συνδεδεμένα RoboCAPs του Προβλήματος 2.	83
Πίνακας 9.27 Πίνακας φερομόνης του Προβλήματος 2.	83
Πίνακας 9.28 Πίνακας πιθανοτήτων του Προβλήματος 2.	84
Πίνακας 9.29 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.....	85
Πίνακας 9.30 Λεπτομέρειες χρηστών του Προβλήματος 3.....	85
Πίνακας 9.31 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.	85
Πίνακας 9.32 Συνδεδεμένα RoboCAPs του Προβλήματος 3.....	86
Πίνακας 9.33 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.....	86
Πίνακας 9.34 Λεπτομέρειες χρηστών του Προβλήματος 3.....	86
Πίνακας 9.35 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.	86
Πίνακας 9.36 Συνδεδεμένα RoboCAPs του Προβλήματος 3.....	86
Πίνακας 9.37 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.....	87
Πίνακας 9.38 Λεπτομέρειες χρηστών του Προβλήματος 3.....	87
Πίνακας 9.39 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.	87
Πίνακας 9.40 Συνδεδεμένα RoboCAPs του Προβλήματος 3.....	87
Πίνακας 9.41 Πίνακας φερομόνης του Προβλήματος 3.	87
Πίνακας 9.42 Πίνακας πιθανοτήτων του Προβλήματος 3.	87
Πίνακας 9.43 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 4.....	89
Πίνακας 9.44 Λεπτομέρειες χρηστών του Προβλήματος 4.....	89
Πίνακας 9.45 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 4.	89
Πίνακας 9.46 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 4.....	90
Πίνακας 9.47 Λεπτομέρειες χρηστών του Προβλήματος 4.....	90
Πίνακας 9.48 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 4.	90

РАНЕЕЗНАМО ПЕРПАА

РАНЕЕЗНАМО ПЕРПАА

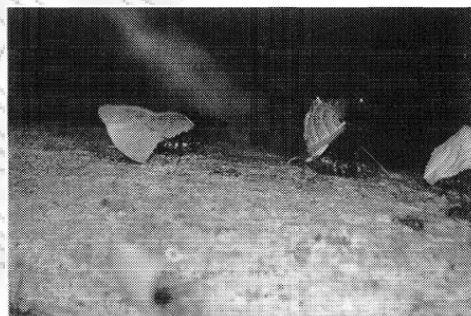
4 Κοινωνικά Έντομα

4.1 Εισαγωγή

Τα έντομα τα οποία ζουν σε αποικίες, όπως τα μυρμήγκια, οι μέλισσες, οι σφήκες¹ και οι τερμίτες, έχουν γοητεύσει τους νατουραλιστές καθώς και τους ποιητές για πολλά χρόνια. «Τι είναι αυτό το οποίο κυβερνά εδώ πέρα; Τι είναι αυτό το οποίο θέτει εντολές, προβλέπει το μέλλον, επεξεργάζεται σχέδια, και διατηρεί την ισορροπία;», έγραψε ο Maeterlinck [1]. Αυτές πράγματι είναι αινιγματικές ερωτήσεις. Κάθε έντομο ξεχωριστά σε μια αποικία κοινωνικών εντόμων φαίνεται να έχει τη δική του ημερήσια διάταξη (agenda), και παρ' όλα αυτά η αποικία φαίνεται τόσο οργανωμένη. Η ολοκλήρωση των ατομικών δραστηριοτήτων δεν φαίνεται να απαιτεί κάποιον επιβλέποντα.

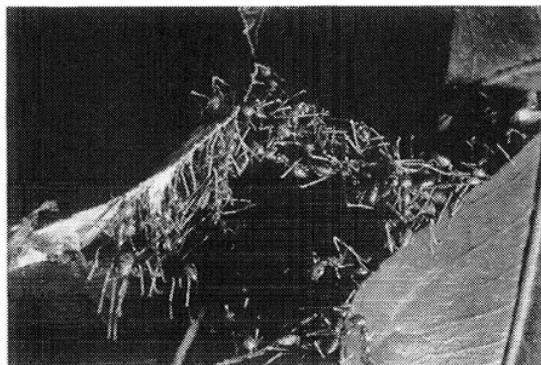
Για παράδειγμα, τα μυρμήγκια *Atta*, κόβουν φύλλα από φυτά και δέντρα για να καλλιεργήσουν μύκητες (Εικόνα 4.1). Οι εργάτες συλλέγουν φύλλα για τροφή εκατοντάδες μέτρα μακριά από τη φωλιά, οργανώνοντας δρόμους κυριολεκτικά από και προς τις τοποθεσίες της τροφής [2].

Οι εργάτες των μυρμηγκιών *Oecophylla*, σχηματίζουν αλυσίδες με τα σώματά τους, που τους επιτρέπει να διασχίσουν μεγάλα κενά καθώς και να τραβήξουν τις άκρες από άκαμπτα φύλλα ώστε να σχηματίσουν μια φωλιά (Εικόνα 4.2). Πολλές αλυσίδες μπορούν να ενωθούν σε μια μεγαλύτερη πάνω στην οποία μπορούν να τρέχουν οι εργάτες. Τέτοιες αλυσίδες δημιουργούν αρκετή δύναμη ώστε να τραβήξουν τις άκρες των φύλλων μαζί. Όταν τα φύλλα είναι στη σωστή θέση, τα μυρμήγκια ενώνουν και τα δύο άκρα με ένα συνεχόμενο νήμα μεταξιού, το οποίο εκκρίνεται από μια ώριμη κάμπια που κρατείται από έναν εργάτη (Εικόνα 4.3) [2][3].

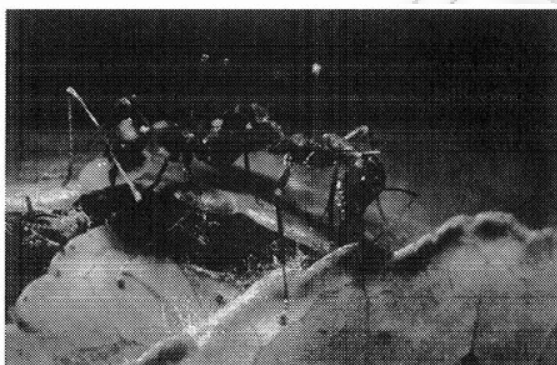


Εικόνα 4.1 Μυρμήγκια *Atta* μεταφέρουν φύλλα πίσω στη φωλιά.

¹ Μόνο ένα τμήμα από τα είδη των μελισσών και των σφηγκών είναι κοινωνικά: τα περισσότερα είναι μοναχικά.



Εικόνα 4.2 Αλυσίδες από μυρμηγκια *Oecophylla*.



Εικόνα 4.3 Δύο εργάτες κρατάνε μια κάμπια στη γνάθο.

Στη φάση κίνησης, τα μυρμηγκια-στρατιώτες (όπως τα *Eciton*), οργανώνουν εντυπωσιακές κυνηγετικές επιδρομές, που περιλαμβάνουν έως και 200.000 εργάτες, κατά τη διάρκεια των οποίων συλλέγουν χιλιάδες θύματα [4][5][6].

Σε μια αποικία κοινωνικών εντόμων, ένας εργάτης δεν εκτελεί συνήθως όλες τις εργασίες, αλλά περισσότερο εξειδικεύεται σε ένα σύνολο εργασιών, ανάλογα με τη μορφολογία του, την ηλικία του ή τυχαία. Αυτός ο διαχωρισμός της δουλειάς, όπου διαφορετικές εργασίες εκτελούνται ταυτόχρονα από ομάδες εξειδικευμένων ατόμων, πιστεύεται ότι είναι πιο ευφυής απ' ό,τι αν οι εργασίες εκτελούνταν διαδοχικά από μη εξειδικευμένα άτομα [7][8].

Σε πολυμορφικά είδη μυρμηγκιών, δύο (ή περισσότερα) φυσικώς διαφορετικά είδη εργατών συνυπάρχουν. Για παράδειγμα, στα είδη *Pheidole*, οι μικροί (*minor*) εργάτες είναι μικρότεροι και διαφέρουν μορφολογικά από τους μεγάλους (*major*) εργάτες. Μικροί και μεγάλοι εργάτες τείνουν να εκτελούν διαφορετικές εργασίες: ενώ οι μεγάλοι κόβουν τη λεία με τη γνάθο τους, οι μικροί τρέφουν τα νεαρά μυρμηγκια ή καθαρίζουν τη φωλιά. Η απομάκρυνση των μικρών εργατών, διεγείρει τους μεγάλους να εκτελούν εργασίες που συνήθως εκτελούνται από τους μικρούς [9]. Αυτή η αντικατάσταση γίνεται μέσα σε δύο ώρες από την απομάκρυνση των μικρών. Πιο γενικά, έχει παρατηρηθεί σε πολλά είδη εντόμων ότι η απομάκρυνση

μιας κλάσης εργατών αντισταθμίζεται από άλλους εργάτες: ο διαχωρισμός της εργασίας επιδεικνύει υψηλό βαθμό πλαστικότητας.

Αν κανένας δεν είναι επικεφαλής πώς μπορούμε να εξηγήσουμε την πολυπλοκότητα και την επιτήδευση της συλλογικής τους παραγωγής; Ένα έντομο είναι ένας πολύπλοκος οργανισμός: μπορεί να χειριστεί πολλές αισθητήριες εισόδους (inputs), να διαφοροποιήσει τη συμπεριφορά του ανάλογα με πολλά ερεθίσματα, συμπεριλαμβάνοντας τις αλληλεπιδράσεις με τα άλλα έντομα της φωλιάς, και να πάρει αποφάσεις με βάση μεγάλες ποσότητες πληροφορίας. Ακόμα, η πολυπλοκότητα του ξεχωριστού εντόμου δεν είναι επαρκής για να εξηγήσει την πολυπλοκότητα του τι μπορεί να κάνει μια αποικία κοινωνικών εντόμων. Πιθανώς, η πιο δύσκολη ερώτηση είναι πώς να συνδέσουμε την ατομική συμπεριφορά με την συλλογική επίδοση. Με άλλα λόγια, πώς προκύπτει η συνεργασία;

Μερικοί από τους μηχανισμούς κάτω από τη συνεργασία καθορίζονται γενετικά: για παράδειγμα, ανατομικές διαφορές μεταξύ ατόμων, όπως οι διαφορές μεταξύ των μικρών και των μεγάλων εργατών στα πολυμορφικά είδη μυρμηγκιών, μπορούν να οργανώσουν το διαχωρισμό της εργασίας. Οι θεωρίες αυτό-οργάνωσης (self-organization – SO) [10][11], που αρχικά αναπτύχθηκαν στα πλαίσια της φυσικής και της χημείας για να περιγράψουν την εμφάνιση μακροσκοπικών μοτίβων από διεργασίες και αλληλεπιδράσεις που ορίζονται σε μικροσκοπικό επίπεδο, μπορούν να επεκταθούν στα κοινωνικά έντομα για να δείξουν ότι η συλλογική συμπεριφορά μπορεί να προκύπτει από αλληλεπιδράσεις μεταξύ των ατόμων που δείχνουν απλή συμπεριφορά: σε αυτές τις περιπτώσεις δεν χρειάζεται να επικαλεστούμε την ατομική πολυπλοκότητα για να εξηγήσουμε την περίπλοκη συλλογική συμπεριφορά. Πρόσφατες έρευνες δείχνουν ότι η αυτό-οργάνωση είναι πράγματι μια μεγάλη συνιστώσα ενός μεγάλου εύρους συλλογικών φαινομένων στα κοινωνικά έντομα [12][16].

Τα μοντέλα που βασίζονται στην SO, δεν αποκλείουν την ατομική πολυπλοκότητα: δείχνουν ότι σε ένα επίπεδο περιγραφής, είναι δυνατόν να εξηγηθεί η πολύπλοκη συλλογική συμπεριφορά, υποθέτοντας ότι τα έντομα είναι απλές αλληλεπιδραστικές οντότητες. Επίσης, ακούγονται λογικά: υποθέτουν ότι μπορεί να είναι δυνατόν να εξηγήσουν κάτι φανερά πολύπλοκο, σε όρους απλών αλληλεπιδραστικών διαδικασιών. Αν μια τέτοια εξήγηση είναι εφικτή, τότε γιατί πρέπει να είναι πιο πολύπλοκη; Μόνο όταν μια τέτοια εξήγηση αποτυγχάνει, πιο πολύπλοκες υποθέσεις θα τεθούν στο μοντέλο [16].

Η ανακάλυψη ότι η SO μπορεί να βρίσκεται σε λειτουργία στα κοινωνικά έντομα, έχει επιπτώσεις όχι μόνο στη μελέτη των κοινωνικών εντόμων, αλλά επίσης μας παρέχει ισχυρά εργαλεία για να μεταφέρουμε τη γνώση σχετικά με τα κοινωνικά έντομα, στον τομέα της σχεδίασης ευφυών συστημάτων. Επομένως, μια αποικία

κοινωνικών εντόμων είναι αναμφισβήτητα ένα αποκεντρωτικό σύστημα λύσης προβλημάτων, που αποτελείται από πολλές σχετικά απλές αλληλεπιδραστικές οντότητες. Τα καθημερινά προβλήματα που λύνονται από μια αποικία, περιλαμβάνουν την εύρεση τροφής, το χτίσιμο ή την επέκταση της φωλιάς, τον ευφυή διαχωρισμό της εργασίας μεταξύ των ατόμων, την ευφυή τροφή των μικρών εντόμων, την ανταπόκριση σε εξωτερικές προκλήσεις, την εξάπλωση ενός συναγερμού, κ.α. Πολλά από αυτά τα προβλήματα έχουν αντιστοιχία με προβλήματα της μηχανικής και της επιστήμης των υπολογιστών. Ένα από τα πιο σημαντικά χαρακτηριστικά των κοινωνικών εντόμων είναι ότι μπορούν να λύσουν αυτά τα προβλήματα με πολύ ευέλικτο και εύρωστο τρόπο: η ευελιξία επιτρέπει προσαρμογή σε μεταβαλλόμενα περιβάλλοντα, ενώ η ευρωστία χαρίζει στην αποικία τη δυνατότητα να λειτουργεί ακόμα κι αν μερικά άτομα αποτύχουν να εκτελέσουν τις εργασίες τους. Τέλος, τα κοινωνικά έντομα έχουν περιορισμένες γνωσιακές ικανότητες: είναι επομένως εύκολο να σχεδιάσουμε αντιπροσώπους (agents), συμπεριλαμβάνοντας και ρομποτικούς αντιπροσώπους, που να μιμούνται τη συμπεριφορά τους σε κάποιο επίπεδο της περιγραφής [16].

Εν ολίγοις, η μοντελοποίηση των κοινωνικών εντόμων σε όρους της SO, μπορεί να βοηθήσει στο σχεδιασμό καταναεμημένων τεχνητών συσκευών επίλυσης προβλημάτων που αυτό-οργανώνονται για να λύσουν προβλήματα – συστήματα με ευφυΐα σμήνους. Πρέπει να πούμε ωστόσο ότι πολύ λίγες εφαρμογές έχουν αναπτυχθεί που να έχουν ευφυΐα σμήνους. Ένας από τους κυριότερους λόγους αυτής της έλλειψης, είναι ότι τα συστήματα με ευφυΐα σμήνους είναι δύσκολο να «προγραμματισθούν», διότι τα βήματα για την επίλυση του προβλήματος δεν είναι προκαθορισμένα αλλά εμφανίζονται σε αυτά τα συστήματα και προκύπτουν από αλληλεπιδράσεις μεταξύ των ατόμων και ανάμεσα στα άτομα και το περιβάλλον, όσο και από τις συμπεριφορές των ίδιων των ατόμων. Έτσι, η χρησιμοποίηση ενός συστήματος με ευφυΐα σμήνους απαιτεί βαθιά γνώση όχι μόνο του πώς πρέπει να εφαρμοστεί η συμπεριφορά των ατόμων, αλλά και τι αλληλεπιδράσεις χρειάζονται για να παραχθεί μια τέτοια συνολική συμπεριφορά.

Μια πιθανή ιδέα είναι να αναπτύξουμε έναν κατάλογο με όλες τις συλλογικές συμπεριφορές που μπορούν να δημιουργηθούν με απλούς αντιπροσώπους που αλληλεπιδρούν: ένας τέτοιος κατάλογος θα ήταν πολύ χρήσιμος για την εγκατάσταση καθαρών συνδέσεων μεταξύ τεχνητών σμηνών και του τι μπορούν να επιτύχουν, αλλά αυτό θα μπορούσε να είναι ένα βαρετό και ατελείωτο επιχείρημα. Ένα πιο λογικό μονοπάτι είναι η μελέτη του πώς τα κοινωνικά έντομα εκτελούν κάποιες συγκεκριμένες εργασίες, μοντελοποιώντας τη συμπεριφορά τους, και η χρήση αυτού του μοντέλου ως βάση για τις τεχνητές παραλλαγές που μπορούν να αναπτυχθούν, είτε με τη ρύθμιση των παραμέτρων του μοντέλου πέρα από το

βιολογικό σχετικό εύρος, είτε με την πρόσθεση μη βιολογικών χαρακτηριστικών στο μοντέλο.

Τον όρο «ευφυΐα του σμήνους» χρησιμοποίησαν πρώτοι οι Beni, Hackwood και Wang [13][14][15], στα πλαίσια των ρομποτικών κυψελωτών συστημάτων, όπου πολλοί απλοί αντιπρόσωποι καταλαμβάνουν περιβάλλοντα μίας ή δύο διαστάσεων ώστε να δημιουργήσουν μοτίβα και να αυτό-οργανωθούν μέσω αλληλεπιδράσεων του κοντινότερου γείτονα (nearest-neighbor interactions). Χρησιμοποιώντας την έκφραση «ευφυΐα του σμήνους» για να περιγράψουμε αυτή μόνο την εργασία φαίνεται άσκοπα περιοριστική: γι' αυτό το λόγο επεκτείνουμε τον ορισμό της ώστε να περιλαμβάνει κάθε προσπάθεια σχεδίασης αλγορίθμων ή κατανεμημένων συσκευών επίλυσης προβλημάτων, εμπνευσμένα από τη συλλογική συμπεριφορά των αποικιών συλλογικών εντόμων και άλλων ζωικών κοινωνιών. Και, όπως περιέργως, αυτός ο ορισμός καλύπτει οριακά τη δουλειά πάνω στα ρομποτικά κυψελωτά δίκτυα, που δεν δανείζεται πολλά από τη συμπεριφορά των κοινωνικών εντόμων.

Από ιστορική σκοπιά, η ιδέα της χρησιμοποίησης συνόλων από αντιπροσώπους ή αυτόματα για την επίλυση προβλημάτων βελτιστοποίησης και ελέγχου γράφων, πλεγμάτων και δικτύων είχε ήδη παρουσιαστεί στις εργασίες των Butrimenko [17], Tsetlin [18], Stefanyuk [19] και Rabin [20]. Τελευταία έγινε εισαγωγή των κινούμενων αυτόματων για την επίλυση προβλημάτων σε γράφους και πλέγματα με την αλληλεπίδραση με τις συνέπειες των προηγούμενων πράξεών τους. Ο Tsetlin [18] αναγνώρισε τα σημαντικά χαρακτηριστικά των αυτόματων που είναι εμπνευσμένα από τη βιολογία, που κάνουν την προσέγγιση που βασίζεται σε σμήνη δυνητικά ισχυρή, τυχαία, αποκεντρωτική, έμμεσες αλληλεπιδράσεις μεταξύ των αντιπροσώπων και αυτό-οργάνωση. Ο Butrimenko [17] εφάρμοσε αυτές τις ιδέες στον έλεγχο των τηλεπικοινωνιακών δικτύων και ο Stefanyuk [19] στη συνεργασία μεταξύ των ράδιο-σταθμών.

4.2 Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization - ACO)

Τα μυρμηγκία έχουν εμπνεύσει αρκετές μεθόδους και τεχνικές, ανάμεσα στις οποίες αυτή που έχει μελετηθεί περισσότερο και είναι η πιο πετυχημένη, είναι η γενικού σκοπού τεχνική βελτιστοποίησης, η οποία είναι γνωστή ως βελτιστοποίηση αποικίας μυρμηγκιών.

Η ACO, εμπνέεται από τη συμπεριφορά συλλογής τροφής ορισμένων ειδών μυρμηγκιών. Αυτά τα μυρμηγκία εναποθέτουν φερομόνη στο έδαφος έτσι ώστε να σηματοδοτούν κάποιο ευνοϊκό μονοπάτι που θα πρέπει να ακολουθήσουν και τα

άλλα μέλη της αποικίας. Η ACO εκμεταλλεύεται έναν παρόμοιο μηχανισμό για την επίλυση προβλημάτων βελτιστοποίησης [20].

Από τις αρχές της δεκαετίας του '90, όταν προτάθηκε ο πρώτος ACO αλγόριθμος, η ACO προσέλκυσε το ενδιαφέρον ενός αριθμού ερευνητών που αυξανόταν, και πολλές πετυχημένες εφαρμογές είναι σήμερα διαθέσιμες. Επιπλέον, μια ουσιαστική συλλογή θεωρητικών αποτελεσμάτων γίνεται διαθέσιμη και προσφέρει σημαντική βοήθεια στους ερευνητές και επαγγελματίες στις επόμενες εφαρμογές της ACO.

4.2.1 Έμπνευση από τη βιολογία

Τις δεκαετίες του '40 και του '50, ο Γάλλος εντομολόγος Pierre-Paul Grassé [21], παρατήρησε ότι ορισμένα είδη τερμιτών αντιδρούν με αυτό που αποκάλεσε «σημαντικό ερέθισμα» (significant stimuli). Παρατήρησε ότι το αποτέλεσμα αυτών των αντιδράσεων μπορεί να δράσει σαν νέο σημαντικό ερέθισμα τόσο για το έντομο που το παρήγαγε όσο και για τα άλλα έντομα της αποικίας. Ο Grassé χρησιμοποίησε τον όρο «stigmergy» [22] για να περιγράψει αυτόν τον συγκεκριμένο τύπο επικοινωνίας στον οποίο οι «εργαζόμενοι παρακινούνται από την απόδοση που έχουν επιτύχει».

Τα δύο κύρια χαρακτηριστικά του stigmergy που το διαφοροποιούν από άλλες μορφές επικοινωνίας είναι τα ακόλουθα:

- Το stigmergy είναι μια έμμεση, μη συμβολική μορφή επικοινωνίας που μεσολαβεί το περιβάλλον: τα έντομα ανταλλάζουν πληροφορίες με το να τροποποιούν το περιβάλλον τους και
- η stigmergy πληροφορία είναι τοπική: πρόσβαση μπορούν να έχουν μόνο τα έντομα που επισκέφτηκαν τον τόπο στον οποίο απελευθερώθηκε (ή την άμεση γειτονιά του).

Παραδείγματα stigmergy μπορούν να παρατηρηθούν στις αποικίες μυρμηγκιών. Σε πολλά είδη μυρμηγκιών, τα μυρμηγκία που περπατάνε προς και από την πηγή της τροφής, εναποθέτουν στο έδαφος μια ουσία που λέγεται φερομόνη. Τα άλλα μυρμηγκία αντιλαμβάνονται την παρουσία της φερομόνης και τείνουν να ακολουθήσουν τις διαδρομές που έχουν την υψηλότερη συγκέντρωση φερομόνης. Με αυτόν τον τρόπο τα μυρμηγκία μπορούν να μεταφέρουν την τροφή στη φωλιά τους με έναν αξιοσημείωτα ευφυή τρόπο.

Οι Deneubourg et al [23], μελέτησαν βαθιά αυτή τη συμπεριφορά των μυρμηγκιών. Σε ένα πείραμα που είναι γνωστό ως «το πείραμα της διπλής γέφυρας», η φωλιά μιας αποικίας μυρμηγκιών Αργεντινής συνδέθηκε με μια πηγή τροφής μέσω δύο γεφυρών ίδιου μήκους (Εικόνα 4.4(α)). Τα μυρμηγκία εξερευνούν τον περιβάλλοντα χώρο της φωλιάς και τελικά να φτάσουν στην πηγή της τροφής.

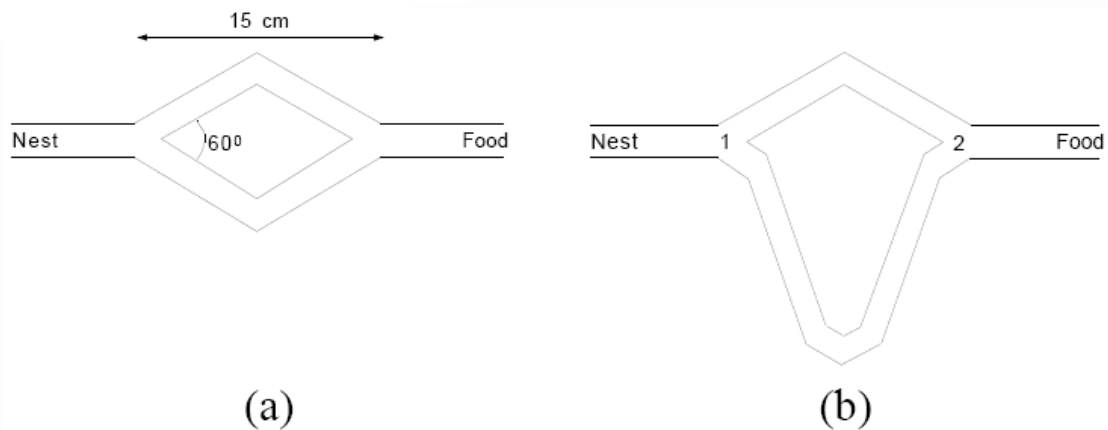
Κατά μήκος τα διαδρομής από τη φωλιά μέχρι την πηγή της τροφής τα μυρμήγκια εναποθέτουν φερομόνη. Αρχικά τα μυρμήγκια επιλέγουν τυχαία μία από τις δύο γέφυρες. Ωστόσο, λόγω τυχαίων διακυμάνσεων, μία από τις δύο γέφυρες παρουσιάζει μεγαλύτερη συγκέντρωση φερομόνης κι έτσι προσελκύει περισσότερα μυρμήγκια. Αυτό έχει ως αποτέλεσμα την ακόμα μεγαλύτερη αύξηση της φερομόνης, και τελικά όλη η αποικία τείνει να χρησιμοποιεί την ίδια γέφυρα².

Αυτή η συμπεριφορά επιπέδου αποικίας, που βασίζεται στην αυτοκατάλυση, δηλαδή στην εκμετάλλευση της θετικής ανατροφοδότησης (feedback), μπορεί να χρησιμοποιηθεί από τα μυρμήγκια για να βρουν τη συντομότερη διαδρομή από την τροφή μέχρι τη φωλιά τους. Οι Goss et al. [24], θεώρησαν μια παραλλαγή του πειράματος της διπλής γέφυρας, στην οποία η μία γέφυρα είναι σημαντικά μακρύτερη από την άλλη (Εικόνα 4.4(β)). Σε αυτήν την περίπτωση, οι τυχαίες διακυμάνσεις στην αρχική επιλογή της γέφυρας μειώνονται πολύ κι ένας δεύτερος μηχανισμός παίζει σημαντικό ρόλο: τα μυρμήγκια που επιλέγουν τυχαία τη μικρότερη γέφυρα επιστρέφουν πρώτα στη φωλιά. Οπότε η μικρή γέφυρα δέχεται φερομόνη πιο νωρίς από την μακριά, κι αυτό το γεγονός αυξάνει την πιθανότητα τα επόμενα μυρμήγκια να την επιλέξουν. Οι Goss et al [24], ανέπτυξαν ένα μοντέλο της παρατηρούμενης συμπεριφοράς: υποθέτοντας ότι μια δεδομένη χρονική στιγμή, m_1 μυρμήγκια έχουν χρησιμοποιήσει την πρώτη γέφυρα και m_2 τη δεύτερη, η πιθανότητα p_1 για ένα μυρμήγκι να επιλέξει την πρώτη γέφυρα είναι:

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \quad (4.1)$$

όπου οι παράμετροι k και h προσαρμόζονται στα δεδομένα του πειράματος, και προφανώς $p_2=1-p_1$. Οι προσομοιώσεις Monte Carlo έδειξαν πολύ καλή προσαρμογή για $k \approx 20$ και $h \approx 2$.

² Ο Deneubourg και οι συνεργάτες του επανέλαβαν το πείραμα αρκετές φορές και παρατήρησαν πως κάθε γέφυρα χρησιμοποιείται στο 50% των περιπτώσεων περίπου.



Εικόνα 4.4 (α) Οι γέφυρες έχουν ίδιο μήκος. (β) Οι γέφυρες έχουν διαφορετικό μήκος.

4.2.2 Η τεχνική της βελτιστοποίησης

Το μοντέλο που παρουσιάστηκε από τον Deneubourg και τους συνεργάτες του για την συμπεριφορά των μυρμηγκιών για την εύρεση τροφής, ήταν η κύρια πηγή έμπνευσης για την ανάπτυξη της βελτιστοποίησης αποικίας μυρμηγκιών. Στην ACO, ένας αριθμός από τεχνητά μυρμηγκία χτίζουν λύσεις για το θεωρούμενο πρόβλημα βελτιστοποίησης, και ανταλλάζουν πληροφορίες για την ποιότητα αυτών των λύσεων μέσω ενός επικοινωνιακού σχήματος που θυμίζει αυτό που έχουν υιοθετήσει τα αληθινά μυρμηγκία.

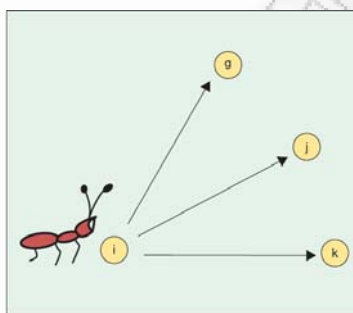
Έχουν προταθεί διαφορετικοί ACO αλγόριθμοι. Ο αυθεντικός ACO αλγόριθμος είναι γνωστός ως Σύστημα Μυρμηγκιών (Ant System) [25] και προτάθηκε στις αρχές της δεκαετίας του '90. Μέχρι τότε είχε προταθεί ένας αριθμός από άλλους ACO αλγόριθμους. Όλοι οι ACO αλγόριθμοι μοιράζονται την ίδια ιδέα, που διευκρινίζεται καλύτερα με ένα παράδειγμα του πώς εφαρμόζονται οι ACO αλγόριθμοι.

4.2.3 ACO για το Πρόβλημα του Πλανόδιου Πωλητή (Traveling Salesman Problem - TSP)

Στο πρόβλημα του πλανόδιου πωλητή, δίνεται ένα σύνολο από πόλεις όπου είναι γνωστές οι μεταξύ τους αποστάσεις. Σκοπός είναι να βρεθεί η συντομότερη διαδρομή που θα επιτρέπει στον πωλητή να επισκεπτει την κάθε πόλη μία και μόνο μία φορά. Σε πιο επίσημη ορολογία, σκοπός είναι να βρεθεί ένα Χαμιλτονιανό μονοπάτι ελάχιστου μήκους, σε έναν πλήρως συνεκτικό γράφο.

Στην ACO, το πρόβλημα αντιμετωπίζεται με την εξομοίωση ενός αριθμού από τεχνητά μυρμηγκία που κινούνται πάνω σε ένα γράφο που κωδικοποιεί το ίδιο το πρόβλημα: κάθε κορυφή αναπαριστά μια πόλη, και κάθε πλευρά αναπαριστά μια σύνδεση μεταξύ δύο πόλεων. Μια μεταβλητή που ονομάζεται φερομόνη σχετίζεται με κάθε πλευρά, και μπορεί να διαβαστεί και να τροποποιηθεί από τα μυρμηγκία [20].

Η βελτιστοποίηση αποικίας μυρμηγκιών, είναι ένας επαναληπτικός αλγόριθμος. Σε κάθε επανάληψη, θεωρείται ένας αριθμός από τεχνητά μυρμήγκια. Κάθε ένα από αυτά χτίζει μία λύση με το να περπατάει από κορυφή σε κορυφή στο γράφο, με τον περιορισμό να μην επισκέπτεται οποιαδήποτε κορυφή έχει ήδη επισκεφτεί στον περίπατό του. Σε κάθε βήμα της κατασκευής της λύσης, ένα μυρμήγκι επιλέγει την επόμενη κορυφή σύμφωνα με το στοχαστικό μηχανισμό που είναι προκατειλημμένος από τη φερομόνη: όταν είναι στην κορυφή i , η επόμενη κορυφή επιλέγεται στοχαστικά μεταξύ των προηγούμενων κορυφών που δεν έχει ακόμα επισκεφτεί (Εικόνα 4.5). Συγκεκριμένα, αν δεν έχει επισκεφτεί ακόμα την κορυφή j , τότε μπορεί να την επιλέξει με πιθανότητα που είναι ανάλογη της φερομόνης που έχει αντιστοιχηθεί στην πλευρά (i,j) [20].



Εικόνα 4.5 Ένα μυρμήγκι στην πόλη i , αποφασίζει την επόμενη πόλη που θα επισκεφτεί μέσω ενός στοχαστικού μηχανισμού: αν δεν έχει επισκεφτεί προηγουμένως την j , τότε μπορεί να την επιλέξει με μια πιθανότητα που είναι ανάλογη της φερομόνης που έχει αντιστοιχηθεί στην πλευρά (i,j) .

Στο τέλος μιας επανάληψης, με βάση την ποιότητα των λύσεων που έχουν κατασκευασθεί από τα μυρμήγκια, οι τιμές της φερομόνης τροποποιούνται προκειμένου να προκαταλάβουν τα μυρμήγκια στις μελλοντικές επαναλήψεις από το να χτίσουν λύσεις που θα είναι παρόμοιες με τις καλύτερες που έχουν κατασκευαστεί ως τώρα.

4.2.4 Η Μεταεвриστική Βελτιστοποίηση Αποικίας Μυρμηγκιών (The Ant Colony Optimization Metaheuristic)

Η ACO έχει τυποποιηθεί σε μια μεταεвриστική για συνδυαστικά προβλήματα βελτιστοποίησης, από τον Dorigo και τους συνεργάτες του [26][27]. Μια μεταεвриστική είναι ένα σύνολο από αλγοριθμικές έννοιες που μπορούν να χρησιμοποιηθούν για τον ορισμό ευριστικών μεθόδων, κατάλληλων για ένα ευρύ σύνολο από διαφορετικά προβλήματα. Με άλλα λόγια, μεταεвриστική είναι ένα αλγοριθμικό πλαίσιο γενικού σκοπού, που μπορεί να εφαρμοστεί σε διαφορετικά προβλήματα βελτιστοποίησης, με σχετικά λίγες τροποποιήσεις. Παραδείγματα μεταεвриστικής είναι η προσομοιωμένη απόπτηση (simulated annealing) [28][29], η tabu search [30]-[32], η iterated local search [33], η evolutionary computation [34]-[37] και η ACO [26][27][38].

Για να εφαρμόσουμε την ACO, σε ένα δεδομένο πρόβλημα βελτιστοποίησης, χρειαζόμαστε ένα επαρκές μοντέλο.

4.2.4.1 Ένα συνδυαστικό πρόβλημα βελτιστοποίησης

Ένα μοντέλο $P = (S, \Omega, f)$ ενός συνδυαστικού προβλήματος βελτιστοποίησης, αποτελείται από:

- ένα χώρο αναζήτησης S που ορίζεται πάνω σε ένα πεπερασμένο σύνολο διακριτών μεταβλητών απόφασης $X_i, i=1, \dots, n$
- ένα σύνολο Ω από περιορισμούς μεταξύ των μεταβλητών και
- μια αντικειμενική συνάρτηση $f: S \rightarrow \mathbb{R}_0^+$ προς ελαχιστοποίηση³

Η γενική μεταβλητή X_i παίρνει τιμές στο $D_i = \{v_i^1, \dots, v_i^{|D_i|}\}$. Μια εφικτή λύση $s \in S$, είναι μια πλήρης ανάθεση τιμών στις μεταβλητές που ικανοποιούν όλους τους περιορισμούς στο Ω . Μια λύση $s^* \in S$, λέγεται καθολικά βέλτιστη, αν και μόνο αν: $f(s^*) \leq f(s) \forall s \in S$.

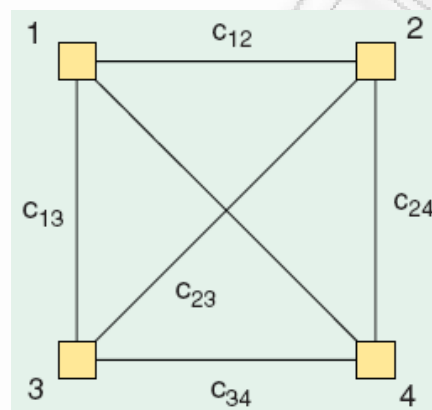
Το μοντέλο ενός συνδυαστικού προβλήματος βελτιστοποίησης χρησιμοποιείται για τον ορισμό του μοντέλου της φερομόνης στην ACO. Μια τιμή φερομόνης σχετίζεται με κάθε πιθανή συνιστώσα των λύσεων, δηλαδή με κάθε πιθανή ανάθεση μιας τιμής σε μια μεταβλητή. Τυπικά, η τιμή της φερομόνης τ_{ij} συσχετίζεται με τη συνιστώσα c_{ij} μιας λύσης, που αποτελείται από την ανάθεση $X_i = v_i^j$. Το σύνολο όλων των πιθανών συνιστωσών των λύσεων συμβολίζεται με C .

Στην ACO, ένα τεχνητό μυρμήγκι κατασκευάζει μια λύση με το να διέρχεται τον πλήρως συνεκτικό γράφο $G_c(V, E)$, όπου V είναι ένα σύνολο από κορυφές και E ένα σύνολο από πλευρές. Αυτός ο γράφος μπορεί να αποκτηθεί από το σύνολο των συνιστωσών των λύσεων C με δύο τρόπους: οι συνιστώσες μπορούν να αναπαρασταθούν είτε από τις κορυφές είτε από τις πλευρές. Τα τεχνητά μυρμήγκια μετακινούνται από κορυφή σε κορυφή, κατά μήκος των πλευρών του γράφου, χτίζοντας σταδιακά μια μερική λύση. Επιπλέον, τα μυρμήγκια εναποθέτουν μια συγκεκριμένη ποσότητα φερομόνης στις συνιστώσες, δηλαδή είτε στις κορυφές είτε στις πλευρές που διέρχονται. Η ποσότητα της φερομόνης $\Delta\tau$ που εναποτίθεται μπορεί να εξαρτάται από την ποιότητα της λύσης που βρέθηκε. Τα επόμενα

³ Οποιοδήποτε πρόβλημα μεγιστοποίησης, μπορεί τετριμμένα να αναχθεί σε πρόβλημα ελαχιστοποίησης: η μεγιστοποίηση μιας συνάρτησης g , είναι ισοδύναμη με την ελαχιστοποίηση της $f=-g$.

μυρμήγκια χρησιμοποιούν την πληροφορία της φερομόνης σαν οδηγό προς υποσχόμενες περιοχές του χώρου αναζήτησης.

Στο TSP, μια λύση μπορεί να αναπαρασταθεί μέσω ενός συνόλου n μεταβλητών, όπου n ο αριθμός των πόλεων. Κάθε μία από αυτές τις μεταβλητές μπορεί να αντιστοιχηθεί με μία πόλη. Η μεταβλητή X_i δείχνει την πόλη που είναι προς επίσκεψη μετά την πόλη i . Εδώ, η συνιστώσα της λύσης $c(i,j)$ δείχνει ότι η υπό ανάλυση λύση ορίζει την πόλη j στην οποία θα πρέπει να γίνει επίσκεψη αμέσως μετά την πόλη i . Σε αυτήν την περίπτωση, ο γράφος της κατασκευής είναι ένας γράφος του οποίου οι κορυφές είναι οι πόλεις του αυθεντικού TSP, και οι πλευρές είναι συνιστώσες της λύσης. Ως αποτέλεσμα, τα μυρμήγκια εναποθέτουν φερομόνη στις πλευρές του γράφου της κατασκευής (Εικόνα 4.6).



Εικόνα 4.6 Εικόνα πιθανού γράφου της κατασκευής για ένα TSP με τέσσερις πόλεις όπου οι συνιστώσες είναι συσχετισμένες με τις πλευρές.

Η μεταεвриστική ACO φαίνεται στον Αλγόριθμο 1. Μετά την αρχικοποίηση, η μεταεвриστική επαναλαμβάνει τρεις φάσεις: σε κάθε επανάληψη, ένας αριθμός λύσεων κατασκευάζεται από τα μυρμήγκια. Αυτές οι λύσεις μετά βελτιώνονται μέσω μιας τοπικής αναζήτησης (local search) (αυτό το βήμα είναι προαιρετικό), και τέλος ανανεώνεται η φερομόνη.

Algorithm 1 The Ant Colony Optimization Metaheuristic

Set parameters, initialize pheromone trails

while termination condition not met **do**

ConstructAntSolutions

ApplyLocalSearch (optional)

UpdatePheromones

endwhile

ConstructAntSolutions: Ένα σύνολο από m τεχνητά μυρμήγκια κατασκευάζει λύσεις από στοιχεία ενός πεπερασμένου συνόλου από διαθέσιμες συνιστώσες λύσεων $C=\{c_{ij}\}$, $i=1,\dots,n$, $j=1,\dots,|D_i|$. Μία κατασκευή της λύσης ξεκινάει από μία κενή μερική λύση $s^p = \emptyset$. Σε κάθε βήμα της κατασκευής, η μερική λύση s^p επεκτείνεται

με την πρόσθεση μιας εφικτής συνιστώσας της λύσης, από το σύνολο $N(s^p) \subseteq C$, που ορίζεται σαν το σύνολο των συνιστωσών που μπορούν να προστεθούν στη μερική λύση s^p χωρίς να παραβιάζουν κάποιον από τους περιορισμούς στο Ω . Η διαδικασία της κατασκευής λύσεων, μπορεί να θεωρηθεί σαν ένας περίπατος πάνω στο γράφο $G_c(V,E)$.

Η επιλογή μιας συνιστώσας των λύσεων από το σύνολο $N(s^p)$, καθοδηγείται από ένα στοχαστικό μηχανισμό που είναι προκατειλημμένος από τη φερομόνη που είναι συσχετισμένη με κάθε στοιχείο του $N(s^p)$. Ο κανόνας για τη στοχαστική επιλογή των συνιστωσών ποικίλει στους διαφορετικούς ACO αλγόριθμους, αλλά σε όλους η έμπνευση προέρχεται από το μοντέλο της συμπεριφοράς των αληθινών μυρμηγκιών που δίνεται στην εξίσωση (4.1).

ApplyLocalSearch: Μόλις έχουν κατασκευαστεί οι λύσεις, και πριν ανανεωθεί η φερομόνη, είναι συνηθισμένο να βελτιώνονται μέσω τοπικής αναζήτησης οι λύσεις που αποκτήθηκαν από τα μυρμηγκία. Αυτή η φάση που εξαρτάται πολύ από το εκάστοτε πρόβλημα, είναι προαιρετική αλλά συνήθως περιλαμβάνεται στους πιο σύγχρονους (state-of-the-art) ACO αλγόριθμους.

UpdatePheromones: Ο σκοπός της ανανέωσης της φερομόνης είναι να αυξήσει τις τιμές της φερομόνης που σχετίζονται με καλές ή υποσχόμενες λύσεις, και να μειώσει αυτές που σχετίζονται με κακές λύσεις. Αυτό συνήθως πετυχαίνεται (i) με την μείωση όλων των τιμών φερομόνης μέσω της εξάτμισης της φερομόνης, και (ii) με την αύξηση των επιπέδων φερομόνης που είναι συσχετισμένα με καλές λύσεις.

4.2.5 Κύριοι ACO αλγόριθμοι

Πολλοί ACO αλγόριθμοι έχουν προταθεί στη βιβλιογραφία. Εδώ θα παρουσιάσουμε το αυθεντικό AS και τις δύο πιο επιτυχημένες παραλλαγές του: το MAX-MIN Σύστημα Μυρμηγκιών (MAX-MIN Ant System) και το Σύστημα Αποικίας Μυρμηγκιών (Ant Colony System). Για να δείξουμε τις διαφορές ανάμεσα στους τρεις αλγόριθμους θα χρησιμοποιήσουμε το TSP.

4.2.5.1 Σύστημα Μυρμηγκιών (Ant System – AS)

Το AS είναι ο πρώτος ACO αλγόριθμος που προτάθηκε στη βιβλιογραφία. Το κύριο χαρακτηριστικό του είναι ότι σε κάθε επανάληψη, οι τιμές της φερομόνης ενημερώνονται από όλα τα m μυρμηγκία που έχουν κατασκευάσει μια λύση. Η φερομόνη τ_{ij} που αντιστοιχεί στην πλευρά που ενώνει τις πόλεις i και j ενημερώνεται ως εξής:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k, \quad (4.2)$$

Όπου ρ είναι ο ρυθμός εξάτμισης, m είναι ο αριθμός των μυρμηγκιών και $\Delta\tau_{ij}^k$ η ποσότητα της φερομόνης που τοποθετήθηκε στην πλευρά (i,j) από το μυρμηγκί k :

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{αν το μυρμηγκί } k \text{ χρησιμοποίησε την πλευρά } (i,j) \text{ στη διαδρομή του} \\ 0, & \text{αλλιώς,} \end{cases} \quad (4.3)$$

όπου Q είναι μια σταθερά, και L_k το μήκος της διαδρομής που κατασκευάστηκε από το μυρμηγκί k .

Στην κατασκευή μιας λύσης, το μυρμηγκί επιλέγει την επόμενη πόλη που θα επισκεφτεί μέσω ενός στοχαστικού μηχανισμού. Όταν το μυρμηγκί k είναι στην πόλη i και έχει κατασκευάσει ως τότε τη μερική λύση s^p , η πιθανότητα να πάει στην πόλη j δίνεται από την σχέση:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{c_{il} \in N(s^p)} \tau_{il}^\alpha \cdot \eta_{il}^\beta}, & \text{αν το } c_{il} \in N(s^p), \\ 0, & \text{αλλιώς,} \end{cases} \quad (4.4)$$

όπου $N(s^p)$ είναι το σύνολο των εφικτών συνιστωσών, δηλαδή οι πλευρές (i,l) , όπου l μια πόλη που δεν την έχει επισκεφτεί ακόμα το μυρμηγκί k . Οι παράμετροι α και β ελέγχουν τη σχετική σημασία της φερομόνης έναντι της μεταευριστικής πληροφορίας η_{ij} , που δίνεται από τη σχέση:

$$\eta_{ij} = \frac{1}{d_{ij}}, \quad (4.5)$$

όπου d_{ij} είναι η απόσταση των πόλεων i και j . Η παράμετρος η_{ij} λέγεται και ορατότητα.

Η χρονική πολυπλοκότητα του AS είναι $O(t \cdot n^2 \cdot m)$, όπου t ο αριθμός των επαναλήψεων που έγιναν. Αν $m=n$, δηλαδή ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων, τότε η πολυπλοκότητα γίνεται $O(t \cdot n^3)$.

4.2.5.2 MAX-MIN Σύστημα Μυρμηγκιών (MAX-MIN Ant System – MMAS)

Αυτός ο αλγόριθμος είναι μια βελτίωση του αυθεντικού AS. Τα χαρακτηριστικά στοιχεία του είναι ότι μόνο τα καλύτερα μυρμηγκία ενημερώνουν τα ίχνη της φερομόνης και ότι η τιμή της φερομόνης είναι φραγμένη. Η ανανέωση της φερομόνης εφαρμόζεται ως εξής:

$$\tau_{ij} \leftarrow \left[(1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{best} \right]_{\tau_{min}}^{\tau_{max}}, \quad (4.6)$$

όπου τ_{max} και τ_{min} τα άνω και κάτω φράγματα που επιβάλλονται στη φερομόνη. Ο τελεστής $[x]_b^a$ ορίζεται ως εξής:

$$[x]_b^a = \begin{cases} a, & \text{αν } x > a \\ b, & \text{αν } x < b \\ x, & \text{αλλιώς} \end{cases} \quad (4.7)$$

και $\Delta\tau_{ij}^{best}$ είναι:

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L_{best}, & \text{αν η } (i, j) \text{ ανήκει στην καλύτερη διαδρομή} \\ 0, & \text{αλλιώς,} \end{cases} \quad (4.8)$$

όπου L_{best} το μήκος της διαδρομής του καλύτερου μυρμηγκιού. Αυτή μπορεί να είναι (υπόκειται στην απόφαση του σχεδιαστή του αλγορίθμου) είτε η καλύτερη διαδρομή που βρέθηκε στην τρέχουσα επανάληψη – iteration best L_{ib} – είτε η καλύτερη λύση που βρέθηκε από την αρχή του αλγορίθμου – best-so-far L_{bs} – είτε ένας συνδυασμός των δύο.

Όσο αφορά τα άνω και κάτω φράγματα τ_{max} και τ_{min} στις τιμές της φερομόνης, τυπικά λαμβάνονται εμπειρικά και ρυθμίζονται ανάλογα με το εκάστοτε πρόβλημα. Ωστόσο παρέχονται οδηγοί για τον ορισμό των τ_{max} και τ_{min} βάσει αναλυτικών μελετών.

Ο αλγόριθμος σε κάθε επανάληψη κατασκευάζει m μονοπάτια, που η κατασκευή του καθενός έχει πολυπλοκότητα $O(n^2)$. Άρα η συνολική πολυπλοκότητα είναι $O(t \cdot n^2 \cdot m)$, όπου t ο αριθμός των επαναλήψεων. Όπως και στο AS, αν $m=n$, δηλαδή ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων, τότε η πολυπλοκότητα γίνεται $O(t \cdot n^3)$.

4.2.5.3 Σύστημα Αποικίας Μυρμηγκιών (Ant Colony System – ACS)

Η πιο σημαντική συμβολή του ACS είναι η εισαγωγή της τοπικής ενημέρωσης της φερομόνης, επιπλέον της ενημέρωσης της φερομόνης που εκτελείται στο τέλος της διεργασίας κατασκευής (καλείται offline ενημέρωση φερομόνης).

Η τοπική ανανέωση της φερομόνης εκτελείται από όλα τα μυρμηγκία μετά από κάθε βήμα κατασκευής. Κάθε μυρμηγκί την εφαρμόζει μόνο στην τελευταία πλευρά που διέσχισε.

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0, \quad (4.9)$$

όπου $\rho \in (0,1]$ είναι ο συντελεστής εξασθένησης της φερομόνης, και τ_0 είναι η αρχική τιμή της φερομόνης.

Ο κύριος σκοπός της τοπικής ανανέωσης είναι να διαφοροποιήσει την αναζήτηση που εκτελείται από τα επόμενα μυρμήγκια κατά τη διάρκεια μιας επανάληψης: με τη μείωση της συγκέντρωσης της φερομόνης στις πλευρές που έχουν ήδη διασχίσει τα μυρμήγκια, τα επόμενα μυρμήγκια ενθαρρύνονται να επιλέξουν άλλες πλευρές, και ως εκ τούτου να παράγουν διαφορετικές λύσεις. Αυτό κάνει λιγότερο πιθανή την περίπτωση αρκετά μυρμήγκια να παράγουν πανομοιότυπες λύσεις κατά τη διάρκεια μιας επανάληψης.

Η offline ενημέρωση της φερομόνης, όμοια με το MMAS, εφαρμόζεται στο τέλος κάθε επανάληψης από ένα μόνο μυρμήγκι, που μπορεί να είναι είτε το καλύτερο της επανάληψης, είτε το καλύτερο έως τώρα. Ωστόσο ο τύπος της ανανέωσης είναι λίγο διαφορετικός:

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}, & \text{αν η } (i,j) \text{ ανήκει στην καλύτερη διαδρομή,} \\ \tau_{ij}, & \text{αλλιώς.} \end{cases} \quad (4.10)$$

Όπως και στο MMAS, $\Delta\tau_{ij} = \frac{1}{L_{best}}$, όπου το L_{best} μπορεί να είναι είτε το L_{ib} είτε το L_{bs} .

Άλλη σημαντική διαφορά μεταξύ του ACS και του AS είναι ο κανόνας της απόφασης των μυρμηγκιών κατά τη διάρκεια της διεργασίας κατασκευής. Στο ACS χρησιμοποιείται ο λεγόμενος ψευδοτυχαίος αναλογικός κανόνας: η πιθανότητα για ένα μυρμήγκι να μετακινηθεί από την πόλη i στην πόλη j εξαρτάται από μια τυχαία μεταβλητή q , ομοιόμορφα κατανομημένη στο διάστημα $[0,1]$, και μια παράμετρο q_0 . Αν $q \leq q_0$, τότε $j = \arg \max_{c_{il} \in N(s^p)} \{ \tau_{il} \cdot \eta_{il}^\beta \}$, αλλιώς χρησιμοποιείται⁴ η εξίσωση (4.4).

⁴ Ο συμβολισμός $\arg \max_x f(x)$ αντιστοιχεί στην τιμή του x για την οποία η $f(\bullet)$ μεγιστοποιείται. Αν το μέγιστο επιτυγχάνεται για πάνω από μία τιμές του x , είναι αδιάφορο ποια απ' όλες θα θεωρηθεί.

РАНЕЕЗНАМО ПЕРПАА

5 Προσομοιωμένη Ανόπτηση (Simulated Annealing)

5.1 Εισαγωγή

Ο Simulated Annealing (Προσομοιωμένη Ανόπτηση) αλγόριθμος παράχθηκε από στατιστικολόγους μηχανικούς. Οι Kickpatrick et al. [39] πρότειναν έναν αλγόριθμο ο οποίος βασίζεται πάνω στην αναλογία μεταξύ της σκλήρυνσης των στερεών και του προβλήματος της λύσης συνδυαστικών προβλημάτων βελτιστοποίησης.

Η διαδικασία ανόπτησης αποτελεί τη φυσική διεργασία της έντονης θέρμανσης (heating up) ενός στερεού και κατόπιν της ψύξης του (cooling down) μέχρι αυτό να κρυσταλλώσει. Τα άτομα μέσα στο υλικό έχουν υψηλές ενέργειες σε υψηλές θερμοκρασίες και επομένως έχουν περισσότερη ελευθερία. Ενώ η θερμοκρασία μειώνεται, μειώνονται παράλληλα και οι ατομικές ενέργειες. Ένας κρύσταλλος με ομαλή δομή επιτυγχάνεται στην κατάσταση όπου το σύστημα έχει ελάχιστη ενέργεια. Εάν η ψύξη γίνει πολύ γρήγορα, διαδικασία γνωστή και ως rapid quenching, εκτεταμένες ανωμαλίες παρατηρούνται στην κρυσταλλική δομή. Το σύστημα δεν φτάνει στην κατάσταση ελάχιστης ενέργειας και καταλήγει σε μια πολυκρυσταλλική δομή η οποία έχει υψηλότερη ενέργεια από την ελάχιστη.

Σε μια δεδομένη θερμοκρασία, η πιθανότητα κατανομής ενεργειών του συστήματος είναι καθορισμένη από την πιθανότητα Boltzmann :

$$P(E) \propto e^{-E/kT} \quad (5.1)$$

Όπου E η ενέργεια του συστήματος, k η σταθερά του Boltzmann, T είναι η θερμοκρασία και $P(E)$ η πιθανότητα το σύστημα να βρίσκεται σε κατάσταση με ενέργεια E .

Σε υψηλές θερμοκρασίες, η $P(E)$ συγκλίνει στο 1 για όλες τις καταστάσεις ενέργειας σύμφωνα με τη σχέση (5.1). Παρατηρούμε επίσης ότι υπάρχει μια μικρή πιθανότητα το σύστημα να έχει υψηλή ενέργεια σε χαμηλές θερμοκρασίες. Επομένως, η στατιστική κατανομή ενεργειών επιτρέπει στο σύστημα να αποδράσει από ένα τοπικό ελάχιστο ενέργειας.

5.2 Βασικά στοιχεία

Στην αναλογία μεταξύ ενός συνδυαστικού προβλήματος βελτιστοποίησης και της annealing διαδικασίας, οι καταστάσεις του στερεού αντιπροσωπεύουν εφικτές λύσεις του προβλήματος βελτιστοποίησης, οι ενέργειες των καταστάσεων αντιστοιχούν στις αξίες της αντικειμενικής συνάρτησης που υπολογίζεται γι' αυτές τις λύσεις, η ελάχιστη ενέργεια αντιστοιχεί στην βέλτιστη λύση του προβλήματος και η rapid quenching μπορεί να θεωρηθεί ως τοπική βελτιστοποίηση.

Ο αλγόριθμος αποτελείται από μία ακολουθία επαναλήψεων. Κάθε επανάληψη αποτελείται από τυχαία αλλαγή της τρέχουσας λύσης ώστε να δημιουργήσει μια νέα λύση μέσα στη γειτονιά της τρέχουσας λύσης. Η γειτονιά είναι ορισμένη από επιλογή του γενετικού μηχανισμού. Μια νέα λύση που δημιουργήθηκε αντιπροσωπεύει μια αλλαγή στη συνάρτηση κόστους που υπολογίστηκε για να αποφασιστεί το κατά πόσον η νέα παραγόμενη λύση μπορεί να γίνει αποδεκτή ως η τρέχουσα λύση. Εάν η αλλαγή στη συνάρτηση κόστους είναι αρνητική η νέα παραγόμενη λύση θεωρείται άμεσα σαν η τρέχουσα λύση. Εναλλακτικά, γίνεται δεκτή σύμφωνα με το κριτήριο Metropolis στηριζόμενο πάνω στην πιθανότητα του Boltzmann.

Σύμφωνα με το κριτήριο Metropolis [40], εάν η διαφορά μεταξύ της συνάρτησης κόστους της τρέχουσας και της νέας παραγόμενης λύσης είναι ίση ή μεγαλύτερη από 0, ένας τυχαίος αριθμός $\delta \in [0,1]$ γεννιέται από μία ομοιόμορφη κατανομή και εάν

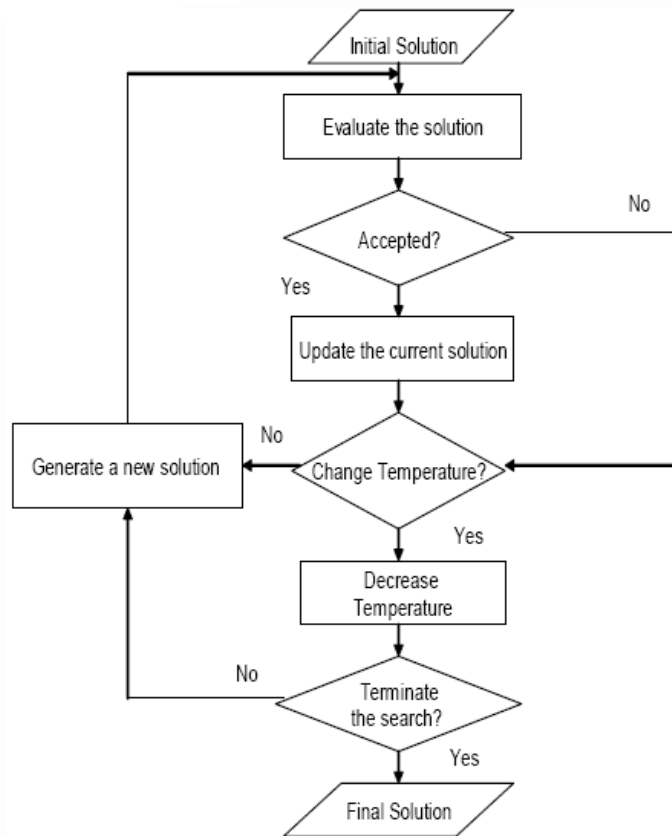
$$\delta \leq e^{-\Delta E/T} \quad (5.2)$$

τότε η νέα παραγόμενη λύση γίνεται αποδεκτή σαν η τρέχουσα λύση. Εάν όχι, τότε η τρέχουσα λύση παραμένει αμετάβλητη.

Το διάγραμμα ροής ενός standard simulated annealing αλγορίθμου παρουσιάζεται στην Εικόνα 5.1. Για να υλοποιήσουμε τον αλγόριθμο για ένα πρόβλημα υπάρχουν τέσσερις θεμελιώδεις επιλογές που πρέπει να γίνουν :

- a. Αναπαράσταση των λύσεων
- b. Καθορισμός της συνάρτησης κόστους
- c. Προσδιορισμός του γενετικού μηχανισμού για τους γείτονες
- d. Σχεδίαση της κρυσταλλικής δομής

Σχεδιάζοντας την κρυσταλλική δομή για έναν simulated annealing αλγόριθμο, τέσσερις παράμετροι πρέπει να οριστούν: Μια αρχική θερμοκρασία, ένας κανόνας ανανέωσης για τη θερμοκρασία, ο αριθμός των επαναλήψεων που εκτελούνται σε κάθε βήμα, και ένα κριτήριο με το οποίο θα σταματάει το ψάξιμο.



Εικόνα 5.1 Διάγραμμα ροής ενός standard simulated annealing αλγόριθμου.

РАНЕЕЗНАМО ПЕРПАА

6 Ad-hoc Δίκτυα

6.1 Ορισμός και ιδιότητες των Ad-hoc Δικτύων

Ένα ad-hoc ασύρματο τηλεπικοινωνιακό δίκτυο αποτελείται από δύο ή περισσότερους κινητούς κόμβους, υπολογιστικές συσκευές (φορητούς υπολογιστές, υπολογιστές χειρός, κινητά τηλέφωνα κ.τ.λ.), οι οποίοι έχουν δυνατότητα για ασύρματη μετάδοση και λήψη δεδομένων. Οι συσκευές μέσα σε ένα τέτοιο δίκτυο έχουν την δυνατότητα επικοινωνίας με οποιαδήποτε άλλη συσκευή, η οποία βρίσκεται στην εμβέλεια τους ή στην εμβέλεια μιας γειτονικής τους συσκευής. Στην πρώτη περίπτωση η επικοινωνία γίνεται απευθείας μεταξύ των δύο κόμβων, ενώ στην δεύτερη περίπτωση η επικοινωνία γίνεται με τη χρήση ενός ή περισσότερων ενδιάμεσων κόμβων, οι οποίοι αναλαμβάνουν την μεταγωγή των δεδομένων από τον αποστολέα στον παραλήπτη.

Ένα ad-hoc ασύρματο τηλεπικοινωνιακό δίκτυο έχει την δυνατότητα να δημιουργείται δυναμικά και αυτόνομα χωρίς να χρειάζεται την παρουσία άλλων ενεργών και μη ενεργών δικτυακών συσκευών. Αυτό σημαίνει ότι έχει την ικανότητα να μεταβάλλει την τοπολογία του καθώς νέοι κόμβοι μπαίνουν σ' αυτό ή αποχωρούν κάποιοι άλλοι. Οι ίδιοι οι κόμβοι αναλαμβάνουν την διαχείριση των πόρων και την επιτέλεση των λειτουργιών του. Ο όρος ad-hoc σημαίνει ότι το δίκτυο μπορεί να πάρει πολλές μορφές, να αποτελείται από κόμβους που κινούνται στο χώρο, να λειτουργεί αυτόνομα και να είναι διασυνδεδεμένο με κάποιο άλλο δίκτυο. Οι συσκευές που μετέχουν σε ένα τέτοιο δίκτυο, πρέπει να μπορούν να αντιλαμβάνονται την παρουσία άλλων συσκευών που θα μπορούσαν να συμμετέχουν στο ίδιο δίκτυο, καθώς και να μπορούν να ενεργοποιήσουν τις κατάλληλες διαδικασίες και πρωτόκολλα διασύνδεσης, ούτως ώστε να είναι αυτό εφικτό, με απώτερο σκοπό την επικοινωνία, την ανταλλαγή δεδομένων και την χρήση των υπηρεσιών του δικτύου.

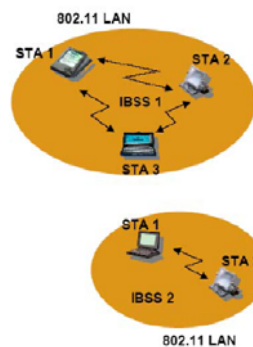
Τα ad-hoc δίκτυα επιτρέπουν την εύκολη ανάπτυξη και λειτουργία τους σε πολύ σύντομο χρονικό διάστημα χωρίς να είναι απαραίτητη η χρήση εξειδικευμένων εφαρμογών και η εκτέλεση διαχειριστικών λειτουργιών ή άλλων ενεργειών από τους χρήστες. Ένα άλλο πλεονέκτημα είναι το γεγονός ότι δεν απαιτείται η χρήση σταθερών δικτυακών υποδομών για τη λειτουργία του δικτύου ενώ η τοπολογία του δικτύου μπορεί να είναι δυναμική [41].

6.2 Λειτουργία των Ad-hoc δικτύων

Τα ασύρματα δίκτυα ad-hoc αποτελούνται από κινητούς κόμβους οι οποίοι πρέπει να επιτελέσουν επιπλέον έργο για να μπορέσει το δίκτυο να λειτουργήσει. Στην περίπτωση που μας αφορά, οι κόμβοι των δικτύων αυτών πρέπει να φροντίσουν να εκτελούνται οι βασικές λειτουργίες ενός δικτύου για τη μεταγωγή

δεδομένων μεταξύ των κόμβων του, εργασία που στα κλασσικά δίκτυα την επιτελούν οι δρομολογητές και τα άλλα ενεργά μη τερματικά στοιχεία του δικτύου. Η καταγραφή των βέλτιστων διαδρομών, μέσω των ενεργών συνδέσεων ενός δικτύου για την μεταφορά δεδομένων, είναι από τις βασικότερες λειτουργίες που πρέπει να έχει ένα δίκτυο υπολογιστών, αφού πολύ απλά χωρίς αυτή δεν είναι δυνατό να υπάρξει.

Υπάρχουν πολλά διαφορετικά είδη πρωτοκόλλων διαθέσιμα σήμερα για την δρομολόγηση δεδομένων σε ένα δίκτυο υπολογιστών, τα οποία μπορούν να λειτουργήσουν αρκετά ικανοποιητικά. Τα πρωτόκολλα αυτά είναι σχεδιασμένα να μπορούν να λειτουργήσουν σε ένα ad-hoc δίκτυο, το οποίο δεν έχει την υποδομή που έχουν τα κλασσικά δίκτυα. Πολλά από τα πρωτόκολλα που έχουμε διαθέσιμα, στα κλασσικά ενσύρματα δίκτυα, δεν μπορούν να λειτουργήσουν σε δίκτυα χωρίς υποδομή. Για να είναι δυνατή η λειτουργία τους, είναι απαραίτητες κάποιες αλλαγές για να μπορέσουν να προσαρμοστούν στα χαρακτηριστικά των ad-hoc δικτύων. Από παρατηρήσεις, πειράματα και προσομοιώσεις που έχουν γίνει, τα πρωτόκολλα αυτά σε καμία περίπτωση δεν μπορούν να έχουν την ίδια απόδοση με αυτή που έχουν όταν εφαρμόζονται στα κλασσικά δίκτυα. Το γεγονός αυτό μας κάνει να πιστεύουμε ότι για να αντιμετωπίσουμε αποτελεσματικά το πρόβλημα της δρομολόγησης χρειαζόμαστε νέα πρωτόκολλα, τα οποία θα δημιουργηθούν για να λειτουργούν αποκλειστικά σε συνθήκες όπως αυτές που υπάρχουν στα ad-hoc δίκτυα [41].



Εικόνα 6.1 Ασύρματο Ad-hoc δίκτυο.

6.3 Εφαρμογές των Ad-hoc Δικτύων

Τα ad-hoc δίκτυα μπορούν να χρησιμοποιηθούν σε αρκετές περιπτώσεις όπου δεν υπάρχει (σταθερή) δικτυακή υποδομή ή όπου αυτή είναι προσωρινά μη διαθέσιμη λόγω βλάβης (π.χ. σε μια φυσική καταστροφή). Μερικές χαρακτηριστικές πιθανές εφαρμογές τους είναι οι εξής:

- **Καταστάσεις έκτακτης ανάγκης.** Σε περιπτώσεις όπου η τοπική δικτυακή υποδομή έχει καταστραφεί ή έχει υποστεί σοβαρή ζημιά (π.χ. λόγω

φωτιάς ή σεισμού), τα ad-hoc δίκτυα μπορούν να χρησιμοποιηθούν ώστε να παρέχουν δικτυακές υπηρεσίες σε ομάδες που τις χρειάζονται, όπως σε ομάδες διάσωσης.

- **Δικτύωση σε εργασία.** Τα ad-hoc δίκτυα μπορούν να χρησιμοποιηθούν για διασύνδεση υπολογιστών σε περιπτώσεις όπως παρακολούθηση συνεδρίων, συναντήσεις για συνεργασία σε κάποιο κοινό έργο κτλ.
- **Δικτύωση σπιτιού.** Οι υπολογιστικές συσκευές που χρησιμοποιούμε (φορητοί υπολογιστές, PDAs, κινητά τηλέφωνα, κτλ) θα μπορούσαν να συνδέονται μέσω ενός ad-hoc δικτύου, έτσι ώστε να είναι πιο εύκολη η προσθήκη και αφαίρεση συσκευών από το δίκτυο.
- **Δίκτυα αισθητήρων.** Για την παρακολούθηση και καταγραφή των περιβαλλοντολογικών συνθηκών σε μια περιοχή, μια λύση θα ήταν η ρίψη (π.χ. από αέρος) ενός μεγάλου αριθμού μικροσκοπικών αισθητήρων, οι οποίοι θα μπορούν να επικοινωνούν ασύρματα μεταξύ για να συγκεντρώνουν τα απαραίτητα δεδομένα. Για παράδειγμα, θα μπορούσαν να χρησιμοποιηθούν σε ένα δάσος με στόχο τον εντοπισμό πυρκαγιάς.

6.4 Χαρακτηριστικά των Ad-hoc δικτύων

Ένα ad-hoc δίκτυο συνήθως αποτελείται από μικρό αριθμό κόμβων κάθε φορά, γεγονός όχι απόλυτο, οι οποίοι μπορεί να εισέρχονται και να εξέρχονται από το δίκτυο με εντελώς τυχαία συχνότητα. Το δίκτυο είναι ετερογενές, δεν αποτελείται δηλαδή από έναν τύπο συσκευών. Μπορεί να αποτελείται από ένα σύνολο PDA, κινητών τηλεφώνων, φορητών υπολογιστών κτλ. τα οποία πρέπει να έχουν δυνατότητα επικοινωνίας μεταξύ τους. Η κατανομή των κόμβων αυτών στο χώρο καθορίζει και την τοπολογία που θα χρησιμοποιηθεί. Αν για παράδειγμα όλες οι συσκευές βρίσκονται πολύ κοντά η μία με την άλλη, είναι εφικτή μία σύνδεση απλού hop από κόμβο σε κόμβο. Αντίθετα αν το δίκτυο εκτείνεται σε μεγάλη γεωγραφική έκταση, απαιτείται multi-hop διασύνδεση μεταξύ των κόμβων. Η σημασία των ad-hoc δικτύων είναι πολύ μεγάλη, κυρίως χάρη στην ευκολία και ταχύτητα με την οποία μπορούν να εγκατασταθούν, αφού δεν απαιτούν την ύπαρξη σταθερής υποδομής. Ένα ακόμα πλεονέκτημα της δυναμικής τους φύσης είναι η εύκολη προσθήκη και απομάκρυνση νέων κόμβων, καθώς και το γεγονός ότι κάθε κόμβος εξαρτάται μόνο από τους γειτονικούς του, με αποτέλεσμα την αυξημένη αξιοπιστία των δικτύων αυτών.

Τα ad-hoc δίκτυα παρουσιάζουν σημαντική ανομοιογένεια, αφού κάθε κόμβος μπορεί να διαφέρει από τους υπόλοιπους σε πολλά χαρακτηριστικά, όπως την υπολογιστική ισχύ, την ακτίνα εκπομπής ή τη διάρκεια ζωής των μπαταριών (αν π.χ.

είναι ένας φορητός υπολογιστής ή ένα PDA). Επιπλέον, τα διάφορα ad-hoc δίκτυα μπορεί να διαφέρουν σε πολλά χαρακτηριστικά τους, όπως τους χρησιμοποιούμενους ρυθμούς επικοινωνίας, στο αν παρέχουν δυνατότητες broadcast ή multicast, στο αν συνυπάρχουν ή όχι με άλλα δίκτυα τα οποία έχουν κάποια σταθερή υποδομή ή τέλος, αν υποστηρίζουν την κινητικότητα των χρηστών και με τι ρυθμούς.

Σημαντικό ρόλο σε κάθε ad-hoc δίκτυο παίζει η ακτίνα μετάδοσης κάθε κόμβου. Συγκεκριμένα, όσο μεγαλύτερη είναι η ακτίνα μετάδοσης των κόμβων, τόσο μικρότερος θα είναι ο μέσος αριθμός μεταδόσεων που θα απαιτείται για την αποστολή ενός πακέτου από ένα κόμβο σε κάποιον άλλο. Από την άλλη μεριά, η μικρή ακτίνα εκπομπής των κόμβων μειώνει την πιθανότητα συγκρούσεων, καθώς και τις παρεμβολές μεταξύ των κόμβων. Με άλλα λόγια, όσο μικρότερη είναι η ακτίνα εκπομπής, τόσο περισσότερες μεταδόσεις θα μπορούν να πραγματοποιούνται ταυτόχρονα. Επιπρόσθετα, η ακτίνα μετάδοσης παίζει καθοριστικό ρόλο και στην κατανάλωση ενέργειας κάθε κόμβου, η οποία είναι μια πολύ σημαντική παράμετρος στα περισσότερα ad-hoc δίκτυα και συχνά η σημαντικότερη στα MANET. Έτσι, η ακτίνα μετάδοσης θα πρέπει να είναι όσο το δυνατό μικρότερη, φροντίζοντας όμως ταυτόχρονα να μην είναι τόσο μικρή που το δίκτυο να παύει να είναι συνεκτικό. Μια καλή επιλογή είναι, συνήθως, να επιλέγεται ακτίνα μετάδοσης, έτσι ώστε κάθε μετάδοση να γίνεται αντιληπτή από περίπου έξι κόμβους.

Οι Micah Adler και Christian Scheideler, προτείνουν ένα μοντέλο τριών επιπέδων για την περιγραφή ενός δικτύου ad-hoc. Αρχικά, έχουμε το επίπεδο Ελέγχου Προσπέλασης Μέσου (Medium Access Control layer), το οποίο είναι υπεύθυνο για την επικοινωνία από σημείο-σε-σημείο (node-to-node) στο φυσικό μέσο. Ακολούθως έχουμε το επίπεδο Επιλογής Διαδρομής, (Route Selection layer), το οποίο είναι υπεύθυνο για την εύρεση κατάλληλων διαδρομών για τα πακέτα. Τέλος, έχουμε το επίπεδο Χρονοπρογραμματισμού (Scheduling layer), που είναι υπεύθυνο για τον καθορισμό της σειράς αποστολής των πακέτων [41].

6.5 Ποιότητα υπηρεσιών στα Ad-hoc δίκτυα

Η ποιότητα των υπηρεσιών που προσφέρει ένα δίκτυο, γίνεται αντιληπτή διαφορετικά από τον κάθε χρήστη. Αυτό είναι άμεσο αποτέλεσμα κυρίως των απαιτήσεων που έχει ο καθένας από τα εργαλεία-εφαρμογές που χρησιμοποιεί. Έτσι, ανάμεσα σε κάποιους χρήστες ενός δικτύου, ο πρώτος μπορεί να έχει απαίτηση για μεγάλο εύρος ζώνης από το δίκτυο-εφαρμογή του, ένας δεύτερος μπορεί να αποζητά πολύ μικρό χρονικό διάστημα από την αποστολή ενός μηνύματος μέχρι τη λήψη του, ενώ ένας τρίτος μπορεί να θέλει επιβεβαίωση ότι κάθε μήνυμα που στέλνει θα φτάσει στον προορισμό του.

Αυτές οι απαιτήσεις είναι πολύ γενικές για να αναλυθούν περαιτέρω. Έχουν ωστόσο κοινό παρονομαστή, την ποιότητα υπηρεσιών.

Είναι γνωστό ότι η ποιότητα υπηρεσιών που μπορεί να προσφέρει ένα δίκτυο σε μία διεργασία εξαρτάται άμεσα από την ποιότητα του δικτύου. Έτσι μπορούν να τεθούν κάποιες πρώτες αρχές:

1. Στις παραμέτρους του δικτύου:
 - a) οι διαθέσιμοι πόροι
 - b) η σταθερότητα των πόρων
2. Το πρωτόκολλο πρέπει να είναι πλήρως προσαρμοζόμενο στις γεωγραφικές αλλαγές του δικτύου, στη μεταβολή των διαθέσιμων πόρων αλλά και στη μικρή χωρητικότητα του δικτύου.

Οπότε είμαστε αναγκασμένοι, για να απλοποιήσουμε το πρόβλημα, να ορίσουμε κάποιες μετρικές ώστε να μπορούμε να έχουμε κάποια μέτρηση της ποιότητας υπηρεσιών που προσφέρεται. Αυτές μπορούν να χωριστούν στις εξής κατηγορίες:

- Μετρικές Επιπέδου Εφαρμογής (ALMs - Application Layer Metrics)
- Μετρικές Επιπέδου Δικτύου (NLMs - Network Layer Metrics)
- Μετρικές επιπέδου MAC (MLMs - MAC Layer Metrics)

Οι MLM και NLM δίνουν μία εκτίμηση της ποιότητας των συνδέσεων, αλλά και τις ικανότητας τους να παράγουν διαδρομές με καλή ποιότητα και σε μικρό χρόνο. Η μετρική ALM διαλέγει το μονοπάτι που είναι πιθανότερο να συναντά τις απαιτήσεις της διεργασίας. Τέλος, θα πρέπει να υπάρχει η δυνατότητα οι προαναφερόμενες μετρικές να μπορούν να προσαρμοστούν, αν αυτό χρειασθεί, σε δυναμικές ανάγκες και αλλαγές του δικτύου.

Μερικές από τις παραμέτρους που θα πρέπει να λαμβάνονται υπόψη από τις μετρικές είναι οι εξής:

1. Για το επίπεδο εφαρμογής
 - a) η συνολική καθυστέρηση που επηρεάζει κυρίως εφαρμογές πραγματικού χρόνου.
 - b) η ικανότητα μεταφοράς δεδομένων για εφαρμογές πολυμέσων.

2. Για το επίπεδο δικτύου

- a) η κατανάλωση ισχύος.
- b) το μέγεθος του καταχωρητή (buffer).
- c) η σταθερότητα του ρυθμού μετάδοσης / λήψης δεδομένων.
- d) η δυνατότητα διόρθωσης λαθών με κώδικα μεταβλητού μήκους.

3. Για το επίπεδο MAC

- a) το SNR (σηματοθορυβικός λόγος).

Ακόμα θα πρέπει να αναφερθεί η δυνατότητα που υπάρχει για την παροχή δυναμικά μεταβαλλόμενης ποιότητας υπηρεσιών (dynamic QoS). Αυτό μπορεί να επιτευχθεί με τη χρήση κατάλληλου υλικού, που πλέον είναι διαθέσιμο.

6.6 Προβλήματα, προκλήσεις

Ένα από τα σημαντικότερα προβλήματα στα ad-hoc δίκτυα είναι η δρομολόγηση. Ο λόγος είναι ότι οι περισσότεροι από τους γνωστούς αλγόριθμους δρομολόγησης έχουν σχεδιαστεί ώστε να λειτουργούν κάτω από συνθήκες οι οποίες είναι πολύ πιο ευνοϊκές από αυτές που ισχύουν σε ασύρματα δίκτυα. Μία από τις βασικότερες ιδιαιτερότητες που πρέπει να αντιμετωπίσουν οι αλγόριθμοι δρομολόγησης των ασύρματων δικτύων είναι η κινητικότητα των χρηστών, η οποία αλλάζει πολύ συχνά την τοπολογία του δικτύου, με αποτέλεσμα να απαιτείται η κατασκευή νέων διαδρομών. Επιπρόσθετα, εξαιτίας του περιορισμένου διαθέσιμου εύρους ζώνης στα ασύρματα δίκτυα, απαιτείται ο αριθμός των σχετικών με την δρομολόγηση μηνυμάτων να είναι περιορισμένος. Επίσης, στα ασύρματα δίκτυα το ποσοστό των πακέτων που χάνονται είναι αρκετά υψηλό, τόσο λόγω της αυξημένης πιθανότητας λαθών μετάδοσης, όσο και της αυξημένης πιθανότητας καταστροφής συνδέσμων (π.χ. εξαιτίας της μετακίνησης ενός κόμβου).

Όλα τα παραπάνω έχουν σαν αποτέλεσμα την διαφοροποίηση σε σχέση με τα ενσύρματα δίκτυα των ιδιοτήτων που επιθυμούμε να έχουν οι αλγόριθμοι δρομολόγησης στα ad-hoc δίκτυα. Έτσι, τα χρησιμοποιούμενα πρωτόκολλα θα πρέπει να είναι κατανεμημένα, με κάθε κόμβο να είναι αρκετά «έξυπνος» ώστε να μπορεί να παίρνει αποφάσεις δρομολόγησης. Αυτό είναι απαραίτητο, αφού ένα κεντρικό πρωτόκολλο δρομολόγησης δεν θα ήταν αξιόπιστο σε περίπτωση κίνησης των κόμβων. Επιπρόσθετα, το πρωτόκολλο θα πρέπει να δημιουργεί γρήγορα δρομολογήσεις, για να μπορούν να χρησιμοποιηθούν πριν αλλάξει η τοπολογία του δικτύου, διατηρώντας παράλληλα το επιπλέον φορτίο στο δίκτυο, για τους σκοπούς της δρομολόγησης, χαμηλό. Εκτός όλων αυτών, το πρωτόκολλο δρομολόγησης είναι επιθυμητό να μπορεί να παίρνει αποφάσεις δρομολόγησης βασιζόμενες και στην

ενεργειακή κατάσταση κάθε κόμβου, καθώς και στην πιθανή επίδραση αυτών των αποφάσεων σε αυτήν. Τέλος, κάθε σύνδεσμος μεταξύ κόμβων θα πρέπει να θεωρείται από το πρωτόκολλο δρομολόγησης ως μίας κατεύθυνσης, αφού η επικοινωνία προς την μία κατεύθυνση μπορεί να περιορίζεται από φυσικούς παράγοντες ή και τη μορφολογία του χώρου, την ακτίνα εκπομπής κάθε κόμβου και άλλα.

Απ' όσα αναφέρθηκαν παραπάνω, η ύπαρξη πληθώρας διαθέσιμων πρωτοκόλλων δρομολόγησης θα πρέπει να είναι αναμενόμενη. Σε γενικές γραμμές μπορούμε να πούμε ότι δεν υπάρχει ένα πρωτόκολλο δρομολόγησης κατάλληλο για την πλειοψηφία των ad-hoc δικτύων, αλλά σε κάθε ad-hoc δίκτυο το πρωτόκολλο δρομολόγησης επιλέγεται με βάση τα ιδιαίτερα χαρακτηριστικά του.

Οι τρέχουσες προκλήσεις που τα ad-hoc ασύρματα δίκτυα καλούνται να αντιμετωπίσουν είναι :

- 1) Multicast,
- 2) QoS υποστήριξη,
- 3) Δρομολόγηση με επίγνωση ισχύος (power aware routing),
- 4) Δρομολόγηση ενίσχυσης θέσης (location aid routing).

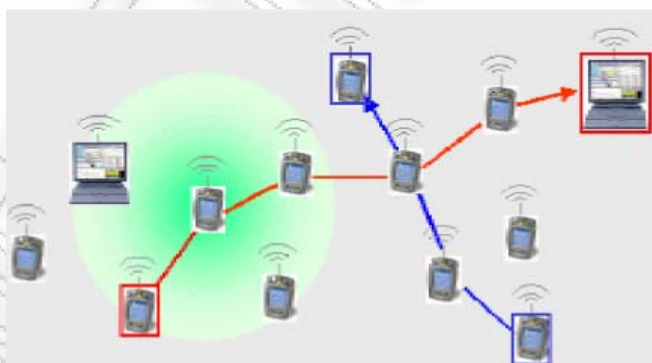
Όπως προαναφέρθηκε, το multicast είναι επιθυμητό ώστε να υποστηρίξει ασύρματες επικοινωνίες πολλαπλών χρηστών. Δεδομένου ότι η ιεραρχία multicast δεν είναι πλέον στατική (η τοπολογία του μεταβάλλεται στο χρόνο), το πρωτόκολλο δρομολόγησης multicast θα πρέπει να μπορεί να ανταπεξέλθει με την κινητικότητα αυτή. Σε όρους QoS, είναι ανεπαρκές να θεωρηθεί το QoS μονάχα στο επίπεδο του δικτύου χωρίς να λάβει υπόψη του το υποκείμενο επίπεδο MAC. Λαμβάνοντας υπόψη τα προβλήματα που συσχετίζονται με τη δυναμική των κόμβων, τα κρυμμένα τερματικά και τα μεταβαλλόμενα χαρακτηριστικά των επικοινωνιακών δεσμών, η υποστήριξη end-to-end QoS είναι θέμα που απαιτεί περισσότερη διερεύνηση. Προς το παρόν, υπάρχει η τάση για μια προσέγγιση ενός προσαρμόσιμου QoS από μια απλή μέθοδο συγκράτησης πόρων με εγγυήσεις QoS. Άλλος σημαντικός παράγοντας είναι η περιορισμένη δυνατότητα παροχής ισχύος των φορητών συσκευών που μπορούν να δημιουργήσουν προβλήματα στην μετάδοση των πακέτων σε ένα κινητό περιβάλλον ad-hoc. Επιπλέον η δρομολόγηση του επικοινωνιακού φόρτου βασιζόμενη στις δυνατότητες ισχύος των κόμβων, είναι ένας τρόπος για το διαχωρισμό των δρομολογητών που έχουν μεγαλύτερη διάρκεια ζωής από κάποιους άλλους. Τέλος αντί της χρήσης αναζήτησης μέσα από broadcast, η δρομολόγηση ενίσχυσης θέσης χρησιμοποιεί πληροφορίες αναφορικά με την

τοποθεσία για να προκαθορίσει σχετιζόμενες περιοχές, ώστε η δρομολόγηση να είναι περιορισμένης έκτασης.

6.7 Mobile Ad-hoc Networks

6.7.1 Εισαγωγή

Ένα κινητό ad-hoc δίκτυο (Mobile Ad-hoc Network - MANET), είναι ένα δίκτυο αυτόνομο και χωρίς σταθερή υποδομή. Μπορεί να αναπτυχθεί απλά και ευέλικτα σχεδόν σε κάθε περιβάλλον, αλλά έχει περιορισμένη ασύρματη κάλυψη και η συνδεσιμότητά του περιορίζεται στα όρια του ίδιου του δικτύου. Η ταχεία ανάπτυξη του διαδικτύου καθώς και των υπηρεσιών και εφαρμογών του και η πορεία των ασύρματων δικτύων τέταρτης γενιάς προς την κατεύθυνση των δικτύων αποκλειστικής χρήσης (All-IP networks), έχουν οδηγήσει σε μια αυξανόμενη απαίτηση για τη δυνατότητα των κόμβων MANET να συνδέονται με το διαδίκτυο και να χρησιμοποιούν τις υπηρεσίες και τις εφαρμογές του. Οι κινητές IP διευθύνσεις και τα πρωτόκολλα κινητών IP επιτρέπουν σε έναν κινητό κόμβο να έχει πρόσβαση στο διαδίκτυο και ν' αλλάζει το σημείο πρόσβασής του χωρίς να χάνει τη σύνδεση. Ο κινητός κόμβος πρέπει να βρίσκεται μέσα στην ακτίνα κάλυψης του σημείου πρόσβασης και να έχει άμεση σύνδεση με αυτό. Έτσι, με τη συνεργασία μεταξύ των πρωτοκόλλων δρομολόγησης του MANET και του πρωτοκόλλου κινητών IP, η συνδεσιμότητα του διαδικτύου με τους κόμβους του δικτύου MANET μπορεί να επιτευχθεί. Πολλές λύσεις έχουν προταθεί για να καταστήσουν τα MANETs ικανά να συνδεθούν με το Διαδίκτυο χρησιμοποιώντας τα πρωτόκολλα κινητών IP [42].



Εικόνα 6.2 Κινητό ad-hoc δίκτυο.

6.7.2 Χαρακτηριστικά των MANETs

Ένα MANET απαρτίζεται από κόμβους (π.χ. ένα δρομολογητή με πολλαπλούς εξυπηρετητές και ασύρματες συσκευές), οι οποίοι κινούνται αυθαίρετα. Οι κόμβοι μπορεί να βρίσκονται πάνω σε αεροπλάνα, πλοία, φορτηγά, ίσως ακόμη και πάνω σε ανθρώπους ή πολύ μικρές συσκευές και μπορεί να υφίστανται πολλαπλοί εξυπηρετητές ανά δρομολογητή. Το σύστημα μπορεί να λειτουργεί απομονωμένα ή διαμέσου πυλών (gateways) και να αλληλεπιδρά με ένα συμβατικό δίκτυο.

Τα MANETs έχουν ορισμένα αξιοπρόσεκτα χαρακτηριστικά όπως:

1. **Δυναμικές τοπολογίες:** Οι κόμβοι δύναται να μετακινούνται αυθαίρετα. Επομένως η τοπολογία του δικτύου, η οποία είναι τυπικά multihop, μπορεί να μεταβληθεί τυχαία και με ταχείς ρυθμούς σε απρόβλεπτους χρόνους και μπορεί να απαρτίζεται ταυτόχρονα από δεσμούς διπλής κατεύθυνσης και ασύμμετρους δεσμούς.
2. **Συνδέσμους με χαμηλό εύρος ζώνης και μεταβαλλόμενη χωρητικότητα:** Οι ασύρματοι σύνδεσμοι θα συνεχίσουν να διατηρούν σημαντικά χαμηλότερη χωρητικότητα από τους αντίστοιχους των συμβατικών δικτύων. Μια επίδραση των σχετικά χαμηλών χωρητικότητων των συνδέσμων, είναι ότι η συμφόρηση αποτελεί περισσότερο τον κανόνα παρά την εξαίρεση, δηλαδή η χωρητικότητα του δικτύου δεν πρόκειται να αγγίξει ή να ξεπεράσει τις συσσωρευμένες απαιτήσεις εφαρμογών. Δεδομένου ότι το κινητό δίκτυο αποτελεί συχνά προέκταση ενός σταθερού συμβατικού δικτύου, οι χρήστες των κινητών δικτύων θα έχουν παρόμοιες απαιτήσεις. Οι απαιτήσεις αυτές θα συνεχίσουν να αυξάνονται καθώς οι αλληλεπιδραστικές εφαρμογές και οι εφαρμογές πολυμέσων θα συνεχίσουν να πληθαίνουν.
3. **Περιορισμούς σε ό,τι αφορά την κατανάλωση ενέργειας:** Μερικοί ή όλοι οι κόμβοι σε ένα MANET μπορεί να στηρίζονται σε μπαταρίες για την παροχή ισχύος. Για τους κόμβους αυτούς, το θέμα της διαχείρισης ενέργειας αποτελεί ένα από τα σημαντικότερα θέματα βελτιστοποίησης κατά το σχεδιασμό του όλου συστήματος.
4. **Περιορισμένη ασφάλεια:** Τα κινητά ασύρματα δίκτυα είναι πιο ευάλωτα σε φυσικές απειλές απ' ό,τι τα συμβατικά ενσύρματα δίκτυα. Θα πρέπει να δοθεί ιδιαίτερη προσοχή σε θέματα υποκλοπών. Οι υφιστάμενες μέθοδοι ασφάλειας συνδέσμων εφαρμόζονται συχνά στα ασύρματα δίκτυα ώστε να ελαχιστοποιηθούν οι απειλές ασφάλειας. Ως πλεονέκτημα μπορεί να αναφερθεί, ότι η αποκεντρωτική φύση της διαχείρισης ενός MANET δικτύου, παρέχει μεγαλύτερη στιβαρότητα σε σχέση με περισσότερο συγκεντρωτικές μεθόδους διαχείρισης.

Επιπλέον, ορισμένα δίκτυα (π.χ. κινητά στρατιωτικά δίκτυα ή δίκτυα ταχείας κυκλοφορίας) μπορεί να είναι ιδιαίτερα μεγάλα (δεκάδες ή εκατοντάδες κόμβοι ανά περιοχή δρομολόγησης). Η ανάγκη για εύκολη κλιμάκωση (scalability), αποτελεί κάτι το δεδομένο. Λαμβάνοντας υπόψη τα προαναφερόμενα χαρακτηριστικά, οι μηχανισμοί ώστε να επιτευχθεί η εύκολη κλιμάκωση θεωρούνται επίσης δεδομένοι.

РАНЕЕЗНАМО ПЕРПАА

7 Γνωσιακά Δίκτυα

7.1 Εισαγωγή

Οι ασύρματες επικοινωνίες προσελκύουν σημαντική ερευνητική προσπάθεια, η οποία αντανακλάται στην πρόοδο της εργασίας που έχει εκτελεστεί σε διεθνή προγράμματα [44], καθώς και σε συζητήσεις σε διεθνή fora [45].

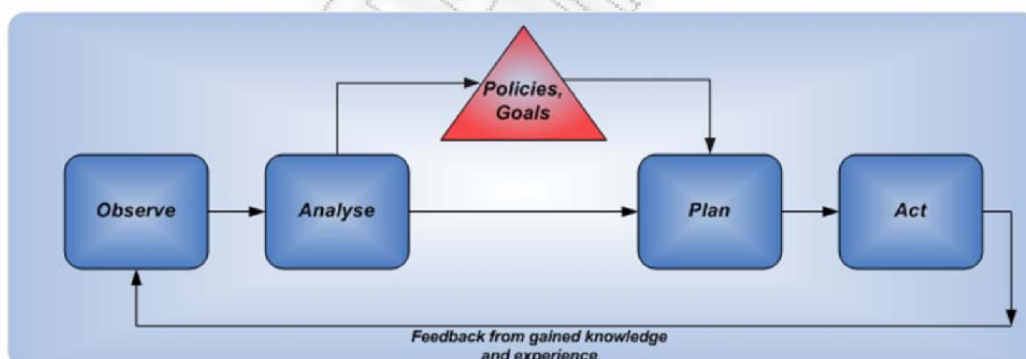
Οι πάροχοι των δικτύων (Network Operators - NOs) έχουν να αντιμετωπίσουν αυξημένη πολυπλοκότητα εξαιτίας της αναπόφευκτης ετερογένειας των δικτύων και των τερματικών υποδομών. Γι' αυτό το λόγο θα πρέπει να αναπτύξουν πολύπλοκες δικτυακές τοπολογίες ετερογενούς φύσης. Οι διαφορετικές Τεχνολογίες Ράδιο-Πρόσβασης (Radio Access Technologies - RATs) θα πρέπει να συνυπάρχουν και να εκμεταλλευτούν συμπληρωματικά (με ευφυή τρόπο). Κάθε RAT έχει διαφορετικές ικανότητες, σε όρους χωρητικότητας, εμβέλειας, φορητότητας, κόστους, κτλ. Οπότε, κάθε RAT είναι καλύτερη για το χειρισμό συγκεκριμένων περιπτώσεων. Σχετικά με αυτό το θέμα, ένας NO θα πρέπει να βασιστεί σε διαφορετικές RATs ώστε να αυξήσει την ικανοποίηση του πελάτη και να επιτύχει το απαιτούμενο επίπεδο QoS με ένα αποτελεσματικό κόστος. Το QoS αναφέρεται στην επίδοση (π.χ. ρυθμός μετάδοσης, καθυστέρηση, κτλ), στη διαθεσιμότητα (π.χ. μικρή blocking probability), στην αξιοπιστία (π.χ. μικρή dropping ή handover probability), καθώς και στην ασφάλεια [46].

Μία μέθοδος για το χειρισμό τέτοιων πολύπλοκων καταστάσεων είναι ο σχεδιασμός μιας πλατφόρμας που θα εκμεταλλεύεται τις δυνατότητες των «γνωσιακών δικτύων» [48][49]. Γενικά, τα γνωσιακά δίκτυα διαθέτουν την ικανότητα να διατηρούν τη γνώση από προηγούμενες αλληλεπιδράσεις με το περιβάλλον και να καθορίζουν τη συμπεριφορά τους σύμφωνα με αυτή τη γνώση όπως και με άλλους στόχους και πολιτικές, έτσι ώστε να προσαρμοστούν στο εξωτερικό ερέθισμα και να βελτιστοποιήσουν την επίδοσή τους. Στην περίπτωση των γνωσιακών δικτύων, αυτός ο ορισμός μπορεί να μεταφραστεί ως η ικανότητα της δυναμικής επιλογής της ρύθμισης του δικτύου, μέσω διαχειριστικής λειτουργικότητας που λαμβάνει υπόψη το πλαίσιο της λειτουργίας (απαιτήσεις του περιβάλλοντος και χαρακτηριστικά), τους στόχους και τις πολιτικές [50] (που αντιστοιχεί σε αρχές), τα προφίλ (ικανότητες), και εκμάθηση των μηχανών [51][52] (για την αντιπροσωπία και διαχείριση γνώσης και εμπειρίας). Η γνωσιακή λειτουργία ως ιδέα μπορεί να επεκταθεί σε πολλά τμήματα ενός επικοινωνιακού συστήματος, π.χ. σε τμήματα του δικτύου, σημεία πρόσβασης ακόμα και σε τερματικά. Άρα η απαιτούμενη διαχειριστική λειτουργικότητα μπορεί να περιλαμβάνει πολλούς συνεργαζόμενους μηχανισμούς, με την έννοια ότι

διαφορετικές εκδόσεις του θα πρέπει να είναι εφαρμόσιμες, οπουδήποτε εισάγεται η γνωσιακή λειτουργία [47].

7.2 Κίνητρο: Γνωσιακά Δίκτυα και η Διαχειριστική Λειτουργικότητά τους

Ο ασύρματος κόσμος έχει ως σκοπό την παροχή αδιάλειπτης φορητότητας στους χρήστες, λαμβάνοντας υπόψη τις ολοένα αυξανόμενες απαιτήσεις των χρηστών. Μεγάλη διευκόλυνση αυτής της τάσης, όπως προαναφέρθηκε, είναι η ιδέα των γνωσιακών δικτύων [53]. Τα γνωσιακά δίκτυα αντικρούουν τα υπάρχοντα, δεδομένου ότι είναι σε θέση να προσαρμόσουν τη λειτουργία τους με το να ανταποκρίνονται σε εξωτερικά ερεθίσματα. Αυτό επιτυγχάνεται με τη διάθεση μηχανισμών που παρακολουθούν εξωτερικές συνθήκες, διατηρούν την πολύτιμη γνώση από αλληλεπιδράσεις με το περιβάλλον και σχεδιάζουν ανάλογα τις μελλοντικές τους δράσεις. Η λειτουργία τους μπορεί να αντανακλασθεί σε ένα βρόχο ανάδρασης (feedback loop) [49], όπως αυτόν της Εικόνα 7.1. Οι βασικές αρχές της γνωσιακής λειτουργίας, προβλέπουν ότι το δίκτυο παρακολουθεί συνεχώς το περιβάλλον, κοιτάζοντας για ενδεχόμενες αλλαγές που μπορούν να επηρεάσουν τη λειτουργία του. Οι παρακολουθήσεις διαμορφώνουν τη βάση για την εκκίνηση machine-learning λογικής, για να δει αν η διαδικασία της αναδιάρθρωσης θα πρέπει να εκτελεστεί. Ο βρόχος επαναλαμβάνεται μέσα σε μια machine-learning διαδικασία, που οδηγεί στη γνώση.



Εικόνα 7.1 Βρόχος ανάδρασης γνωσιακών δικτύων.

Σε αυτό το σημείο μπορεί να προκύψει μια λογική ερώτηση: «Ποιος είναι ο βέλτιστος τρόπος διαχείρισης των διάφορων οντοτήτων που σχηματίζουν ένα γνωστικό δίκτυο»; Η απάντηση σε αυτό το ερώτημα μπορεί να είναι πολύπλοκη. Συγκεκριμένα, τα δίκτυα ράδιο-πρόσβασης έχουν κλασικά σχεδιαστεί και αναπτυχθεί για να καλύψουν τις απαιτήσεις της κίνησης των προγραμματισμένων υπηρεσιών με μία στατική προσέγγιση και από μέσα χειρωνακτικής ρύθμισης των στοιχείων του δικτύου, λαμβάνοντας υπόψη την κίνηση σε ώρες αιχμής σε κάθε γεωγραφική ζώνη. Ωστόσο, η συνεχόμενη αύξηση των απαιτήσεων αύξησε επίσης την ανάγκη για την ανάπτυξη νέων τεχνολογιών και δικτύων που πρέπει να είναι

βέλτιστα σχεδιασμένα και διαχειρισμένα με το να διαλέγουν ανάμεσα στις επιλογές εύρεσης νέων τοποθεσιών, συν-εντοπισμό τοποθεσιών ή μετακίνησης σε αναδιάρθρωμένους πομποδέκτες. Επίσης, στην περίπτωση των γνωσιακών δικτύων, η καινούρια λειτουργικότητα θα πρέπει να σχεδιάζει και να διαχειρίζεται αποτελεσματικά ένα συνεχώς μεταβαλλόμενο δίκτυο, δεδομένου ότι θα πρέπει να προσαρμοστεί στις εξωτερικές απαιτήσεις οι οποίες επίσης αλλάζουν πάνω στο χρόνο και το χώρο.

Επιπλέον, δεδομένου ότι ένα γνωστικό δίκτυο αποτελείται από πολλά στοιχεία και τερματικά με υψηλή φύση ετερογένειας, τοποθετημένα σε διαφορετικές θέσεις, μια προσέγγιση κεντρικής διαχείρισης γίνεται απαγορευτικά πολύπλοκη και ακατάλληλη. Ως εκ τούτου, προσεγγίσεις κατανεμημένης διαχείρισης, που βασίζονται σε συναφείς τεχνολογίες, π.χ. αυτόνομη υπολογιστική, είναι σήμερα στο επίκεντρο. Αυτή η προσέγγιση μπορεί να προσφέρει επεκτασιμότητα, σταθερότητα και διαμορφοποίηση (modularity) (που παρέχει χαμηλή πολυπλοκότητα).

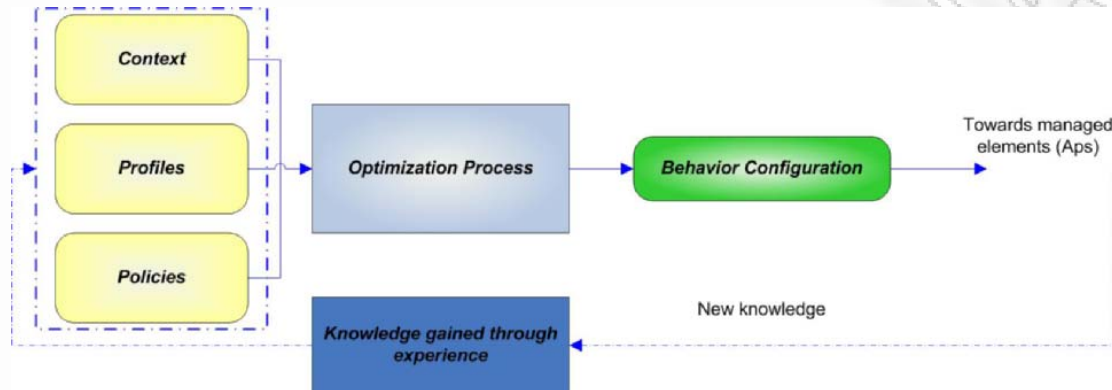
7.3 Περιγραφή του προβλήματος

Από το γενικό ορισμό των γνωσιακών δικτύων συνεπάγονται μερικές πολύ αναπτυγμένες ικανότητες, που πηγάζουν από την ανάγκη να περικλείονται χαρακτηριστικά αναδιάρθρωσης (αλλαγή στη συμπεριφορά του τμήματος, αντανάκλαση στις παραμέτρους/παραλλαγές υποδομών), βελτιωμένες με γνωσιακές δυνατότητες. Σαν μέρος της αναδιάρθρωσης, στα φυσικά και MAC επίπεδα, μπορεί να υπάρξουν στοιχεία (συστατικό υλικού, όπως αναδιάρθρωσιμοι πομποδέκτες) που αλλάζουν δυναμικά τις RATs στις οποίες λειτουργούν και το φάσμα που χρησιμοποιούν, ώστε να βελτιώσουν τα προσφερόμενα επίπεδα QoS. Από την άλλη πλευρά, τέτοιες αλλαγές θα πρέπει να εκτελούνται με τον καλύτερο δυνατό τρόπο, π.χ. οι αλλαγές θα πρέπει να βασίζονται στις εφαρμόσιμες πολιτικές που σχετίζονται με το πλαίσιο (συνδυασμός από περιβαλλοντικά χαρακτηριστικά και συγκεκριμένες περιπτώσεις απαιτήσεων) και στην επιλογή του πιο κατάλληλου αναδιάρθρωμένου μοτίβου. Γενικά αυτή η ημί-κατανεμημένη προσέγγιση καλύπτει το κίνητρο, τις απαιτήσεις, τη λειτουργικότητα και τις μηχανικές προκλήσεις για μια κατανεμημένη λειτουργικότητα, που αποφέρει ισχυρά και επεκτάσιμα μέσα που οδηγούν σε γνωσιακά, ασύρματης πρόσβασης, δίκτυα. Επιπλέον, περιλαμβάνει την ανάπτυξη καινούριων μηχανισμών για την αυτοματοποίηση της διαδικασίας απόφασης της βέλτιστης επαναρύθμισης, άρα διευκόλυνση και βελτιστοποίηση των μηχανισμών σχεδίασης και διαχείρισης.

Η Εικόνα 7.2 παρέχει τη γενική περιγραφή της λειτουργικότητας διαχείρισης που προτάθηκε για τη διαχείριση ενός τμήματος του γνωστικού δικτύου.

Οι προτεινόμενοι μηχανισμοί διαχείρισης αναλαμβάνουν αποφάσεις που επηρεάζουν τη στοίβα πρωτοκόλλων με ένα τρόπο διασταύρωσης επιπέδων. Πιο

κάτω παρατίθενται πληροφορίες για τις εισόδους του προβλήματος (πλαίσιο, προφίλ και πολιτικές) και την έξοδο (διαμόρφωση και συμπεριφορά), όπως και για τα απαραίτητα γνωσιακά χαρακτηριστικά που καλύπτει η λειτουργικότητα.



Εικόνα 7.2 Λειτουργικότητα διαχείρισης για τμήματα γνωστικού δικτύου.

7.3.1 Πλαίσιο

Αυτό το τμήμα της εισόδου είναι πιθανότατα το πιο κρίσιμο, αναφερόμενοι στην ουσία του προβλήματος, λόγω των αλληλεπιδράσεών του με το περιβάλλον, που αποτελεί τους βασικούς λόγους που παρακινούν ένα σύστημα να προσαρμοστεί στα ερεθίσματα. Η πληροφορία από το πλαίσιο παρακολουθείται και ανιχνεύεται για κάθε στοιχείο του δικτύου και για το περιβάλλον του. Η πληροφορία του πλαισίου αποκαλύπτει την κατάσταση των στοιχείων, στο τμήμα του δικτύου, και του περιβάλλοντός τους. Το πλαίσιο περιλαμβάνει πληροφορίες για τις τοποθεσίες, την ώρα, την απαίτηση κίνησης, τα επίπεδα φορητότητας, τις συνθήκες παρεμβολής με διαχειρισμένα στοιχεία, κτλ. [47].

7.3.2 Προφίλ

Αυτό το τμήμα παρέχει πληροφορία για τις υποψήφιας διαμορφώσεις των στοιχείων του τμήματος, όπως το σύνολο των πομποδεκτών του κάθε στοιχείου, το σύνολο των λειτουργικών RATs, όπως και το σύνολο των φερόντων φάσματος (spectrum carriers). Επίσης, αυτό το τμήμα περιγράφει τα προφίλ (π.χ. προτιμήσεις, απαιτήσεις, περιορισμούς) των κλάσεων των χρηστών, των εφαρμογών, των τερματικών, κτλ. [47].

7.3.3 Πολιτικές

Οι πολιτικές ορίζουν κανόνες και λειτουργικότητα που θα πρέπει να ακολουθηθούν στο χειρισμό του πλαισίου. Συγκεκριμένα, αυτό το τμήμα παρέχει πληροφορίες για τις ΟΧΙ πολιτικές σε σχέση με στρατηγικές αναδιάρθρωσης, δηλαδή ΟΧΙ προτιμήσεις και προτεραιότητες σε στόχους που πρέπει να επιτευχθούν. Αυτά σχετίζονται με τη μεγιστοποίηση των επιπέδων QoS (επιδόσεις,

διαθεσιμότητα, αξιοπιστία), και την ελαχιστοποίηση των παραγόντων κόστους (κατανάλωση πόρων). Επιπλέον αυτό το τμήμα παρέχει πληροφορίες για τις ΟΧΙ συμφωνίες με συνεργαζόμενα ΟΧΙ. Πολιτικές και στόχοι θα πρέπει να εγγυώνται για το μέγιστο δυνατό επίπεδο σταθερότητας.

7.3.4 Έξοδος

Η έξοδος παρέχει δράσεις που θα καθορίσουν τη συμπεριφορά των τμημάτων του δικτύου (διαμορφώσεις σε όλα τα επίπεδα της στοίβας πρωτοκόλλων για κάθε στοιχείο του τμήματος).

Στο φυσικό και MAC επίπεδο θα υπάρχει η επιλογή των RATs, του φάσματος, της ισχύος της μετάδοσης, όπως και άλλες παράμετροι και αλγόριθμοι ανάλογα με το στοιχείο του δικτύου και τις ικανότητες των τερματικών.

Στο επίπεδο δικτύου θα υπάρχουν δράσεις που σχετίζονται με τη διασύνδεση των στοιχείων (τρέχουσες ή αναδυόμενες τοπολογίες δικτύου όπως πλέγμα), έλεγχο δρομολόγησης και συμφόρησης.

Στο επίπεδο εφαρμογής, θα υπάρχουν δράσεις που σχετίζονται με την κατανομή των εφαρμογών σε επίπεδα QoS.

7.3.5 Γνωσιακά Χαρακτηριστικά

Η λειτουργικότητα διαχείρισης για τμήματα γνωστικού δικτύου αποτελεί ένα σημαντικό βήμα προς συμπεριφορές σχεδίασης και διαχείρισης τμημάτων καθαρά γνωστικού δικτύου. Αυτό απεικονίζεται στην είσοδό του, στην έξοδό του, και στις τεχνικές βελτιστοποίησης. Συγκεκριμένα, η προτεινόμενη λειτουργικότητα είναι εφαρμόσιμη σε διάφορα επίπεδα διανομής, κυμαινόμενα από ημί-κατανεμημένα σε πλήρως κατανεμημένα σχήματα. Οι αλγόριθμοι επίλυσης γι' αυτό το τμήμα της γενικής λειτουργικότητας διαχείρισης για γνωσιακά δίκτυα, μπορούν να βασιστούν σε τεχνικές βελτιστοποίησης, επιπρόσθετα της λειτουργικότητας εκμάθησης μηχανής και των τεχνικών τεχνητής ευφυΐας. Στην πραγματικότητα, η λειτουργικότητα εκμάθησης μηχανής και οι αλγόριθμοι τεχνητής ευφυΐας αποδίδουν τη λειτουργικότητα διαχείρισης με γνώση και εμπειρία (τουλάχιστον) στις περιοχές που περιγράφονται παρακάτω και μπορούν να τροποποιήσουν τη λειτουργικότητα στο γνωστικό τομέα.

Η πληροφορία του πλαισίου λαμβάνεται μέσω αλληλεπιδράσεων με το περιβάλλον, που οδηγεί σε λογική και αντίληψη, μέσω κατάλληλων τεχνικών εκμάθησης μηχανής. Έτσι, το τμήμα δικτύου είναι ικανό να κερδίσει γνώση από αυτές τις αλληλεπιδράσεις και να γνωρίζει τη βέλτιστη συμπεριφορά για διάφορα πλαίσια.

Οι σταθερές ενημερώσεις στην πληροφορία που παρέχεται από το τμήμα των «προφίλ» της λειτουργικότητας οδηγεί σε σημαντικά κέρδη γνώσης σε σχέση με τη συμπεριφορά του χρήστη. Αυτό είναι σημαντικό στην αναζήτηση για αδιάλειπτη τροφοδοσία υπηρεσιών και για απaráμιλλη ποιότητα, προσαρμοσμένες στις ανάγκες κάθε χρήστη ξεχωριστά. Από αυτό συνεπάγεται ότι η διεργασία εξυπηρέτησης των χρηστών διευκολύνεται και βελτιστοποιείται μέσω εμπειρίας.

Γνωσιακά χαρακτηριστικά βρίσκονται επίσης στο τμήμα «πολιτικών» της λειτουργικότητας διαχείρισης. Συγκεκριμένα, η καταλληλότητα και η αποδοτικότητα διαφορετικών πολιτικών κατέχει μεγάλη σημασία στο χειρισμό πολύπλευρων συναφών καταστάσεων. Συνεπώς, η εκμάθηση της βέλτιστης πολιτικής και του πιο κατάλληλου στόχου, μπορεί να γίνει πολύτιμη για τα ΟΧΙ στον επιτυχημένο χειρισμό δύσκολων προϋποθέσεων [47].

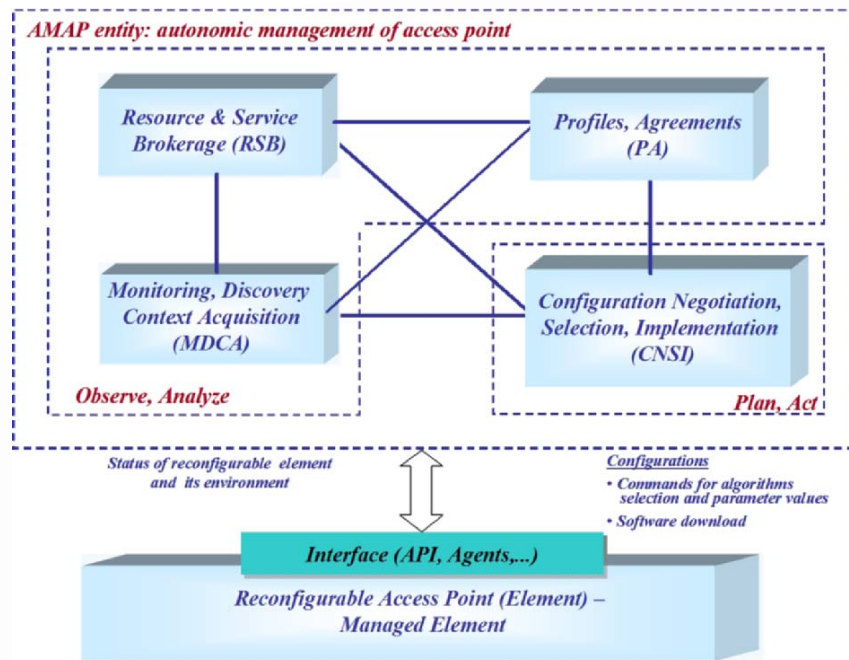
7.4 Λειτουργικότητα Διαχείρισης για Γνωσιακά Σημεία Πρόσβασης

Αυτή η ενότητα ασχολείται με τη λειτουργικότητα διαχείρισης που ενεργεί σε ένα πλήρως κατανεμημένο τρόπο διαμόρφωσης και είναι εφαρμόσιμη στα σημεία πρόσβασης μιας γνωσιακής υποδομής. Η γενική περιγραφή του προβλήματος στην επίλυση του οποίου στοχεύει αυτή η λειτουργικότητα παραμένει αμετάβλητη σε σχέση με τη διαχείριση ολόκληρων τμημάτων δικτύου, δηλαδή με βάση πλαίσιο, προφίλ και πληροφορία πολιτικών, στοχεύει στην απόφαση σχετικά με την πιο κατάλληλη συμπεριφορά ενός σημείου πρόσβασης της υποδομής (ενός μεμονωμένου στοιχείου). Παρακάτω περιγράφεται μια δυνατή λύση στο πρόβλημα, δηλαδή η Αυτόνομη Διαχείριση Σημείων Πρόσβασης (Autonomic Management of Access Points – AMAP), όπως και τα χαρακτηριστικά που ενεργοποιούν και υποστηρίζουν γνωσιακή λειτουργία.

7.4.1 Αυτόνομη Διαχείριση Σημείων Πρόσβασης (Autonomic Management of Access Points – AMAP)

7.4.1.1 Εισαγωγή και Λειτουργική Ανάλυση

Η αφετηρία για την λειτουργική ανάλυση της προτεινόμενης λειτουργικότητας διαχείρισης είναι η εισαγωγή των συστατικών μιας λειτουργικής οντότητας AMAP. Η Εικόνα 7.3 δείχνει αυτά τα συστατικά. Η λειτουργική αρχιτεκτονική AMAP περιλαμβάνει τμήματα που παρέχουν: (i) προφίλ και συμφωνίες (Profiles and Agreements - PA), (ii) παρακολούθηση, ανίχνευση και απόκτηση πλαισίου (Monitoring, Discovery and Context Acquisition - MDCA), (iii) μεσιτεία πόρων και υπηρεσιών (Resource and Service Brokerage – RSB), (iv) διαπραγμάτευση διαμόρφωσης, επιλογή και εκτέλεση (Configuration Negotiation, Selection and Implementation – CNSI), με βάση πολιτικές, στόχους προφίλ και πληροφορία πλαισίου.



Εικόνα 7.3 Λειτουργική Ανάλυση μιας AMAP οντότητας.

7.4.1.2 Προφίλ και Συμφωνίες

Το συστατικό θα πρέπει να παρέχει πληροφορία για χρήστες, για τις δυνατότητες του εξοπλισμού που παρέχεται από τα σημεία πρόσβασης, όπως και για τις δυνατότητες του στοιχείου (σημείο πρόσβασης).

- **Διοικούμενα σημεία πρόσβασης.** Το τμήμα των σημείων πρόσβασης περιγράφει όλες τις δυνατές διαμορφώσεις κάθε πομποδέκτη, συνεπώς και η διαμόρφωση του σημείου πρόσβασης.
- **Εξοπλισμός.** Αυτό το τμήμα του συστατικού δείχνει τις ικανότητες του εξοπλισμού στην περιοχή εξυπηρέτησης, με ένα τρόπο ανάλογο με αυτόν των διοικούμενων στοιχείων.
- **Τελικοί χρήστες.** Η πληροφορία θα πρέπει να διευκρινίζει εφαρμογές, επιτρεπτά επίπεδα QoS, τη σημαντικότητα (χρησιμότητα), κόστος (επιτρεπτό ή ανεκτό), κτλ.

7.4.1.3 Παρακολούθηση, ανίχνευση και απόκτηση πλαισίου

Το συστατικό θα πρέπει να παρέχει τα μέσα για: (i) *απόκτηση του πλαισίου* στο οποίο τα διοικούμενα σημεία πρόσβασης λειτουργούν, (ii) *παρακολούθηση* της αποδοτικότητας με την οποία χειρίζεται κάθε κατάσταση του πλαισίου, (iii) *ανίχνευση* εναλλακτικών που μπορούν να χρησιμοποιηθούν για χειρισμό πλαισίου.

- **Απόκτηση πλαισίου.** Το πλαίσιο, στο οποίο τα διοικούμενα στοιχεία λειτουργούν, αποτελείται από: (i) την τοποθεσία, (ii) την εποχή του

χρόνου, (iii) την κίνηση, την φορητότητα και τις συνθήκες παρεμβολής που συναντώνται και προκαλούνται από τον εξοπλισμό/χρήστες στην περιοχή εξυπηρέτησης.

- *Παρακολούθηση.* Αυτή η διαδικασία συλλέγει βασικά δεδομένα από του πομποδέκτες του διοικούμενου στοιχείου. Τα βασικά δεδομένα που συλλέχθηκαν από πομποδέκτες μπορούν να κατεργασθούν για την παραγωγή συνολικών δεδομένων σε διάφορα επίπεδα αφαιρέσεων.
- *Ανίχνευση.* Αυτή η λειτουργικότητα αυτού του τμήματος βρίσκεται στην εκτίμηση της εκτέλεσης εναλλακτικών διαμορφώσεων σε ένα συγκεκριμένο πλαίσιο. Επομένως, αυτό το τμήμα του συστατικού θα πρέπει να έχει πληροφορία για την επιτεύξιμη εκτέλεση, δηλαδή επίπεδα παρεμβολής, ρυθμός μετάδοσης, κάλυψη, συσχετισμένη με κάθε εναλλακτική διαμόρφωση πομποδέκτη.

7.4.1.4 Μεσιτεία Πόρων και Υπηρεσιών

Αυτό το συστατικό επιτρέπει σε μια AMAP οντότητα να αλληλεπιδρά με άλλες οντότητες του περιβάλλοντός της, έτσι ώστε να αποκτήσει δεδομένα για την κατάσταση των «γειτονικών» στοιχείων. Τελικά, η ανάγκη γι' αυτές τις αλληλεπιδράσεις μπορεί να ελαττωθεί, και κάθε σημείο πρόσβασης να μπορεί να αναπτύξει γνώση για τις ικανότητες των κοντινών του στοιχείων. Ωστόσο, η ύπαρξη μιας τέτοιας οντότητας μπορεί να διασφαλίσει την ομαλή μετάβαση από τα τρέχοντα μοντέλα διαχείρισης στα αυτόνομα.

7.4.1.5 Διαπραγμάτευση Διαμόρφωσης, Επιλογή και Εκτέλεση

Ο ρόλος αυτού του συστατικού είναι να εξετάζει τις εισόδους και να αποφασίζει τη βέλτιστη συμπεριφορά του διοικούμενου σημείου πρόσβασης. Η AMAP οντότητα λαμβάνει υπόψη πολιτικές, που καθορίζουν κανόνες (περιορισμούς) που θα πρέπει να ικανοποιηθούν, και λειτουργικότητα που θα πρέπει να ακολουθηθεί για την επίτευξη των στόχων, που έχει οριστεί από τις πολιτικές. Λαμβάνοντας υπόψη τα παραπάνω, διαφορετικά πλαίσια μπορούν να αντιμετωπιστούν, σύμφωνα με μια συγκεκριμένη στρατηγική, μέσω της εκτέλεσης των αποφάσεων που πάρθηκαν.

7.4.1.6 Γνωσιακά Χαρακτηριστικά

Σε αυτή την ενότητα γίνεται συζήτηση σχετικά με τα γνωσιακά χαρακτηριστικά που περιλαμβάνει η λειτουργικότητα διαχείρισης για γνωσιακά σημεία πρόσβασης. Αυτά τα χαρακτηριστικά χωρίζονται στα διαφορετικά AMAP συστατικά όπως περιγράφηκε πιο πάνω.

Το τμήμα «προφίλ και συμφωνίες» της λειτουργικότητας δρα σε ένα γνωστικό περιβάλλον με το να αναπτύσσει γνώση για το προφίλ χρήστη, σχετικά με τη συμπεριφορά ενός συγκεκριμένου χρήστη, σχετικές εφαρμογές, κίνηση και

φορητότητα, σε μια δεδομένη χρονική εποχή/ρόλο-χρήστη, τοποθεσία, και τις προτιμήσεις του χρήστη για επίπεδα QoS σε μια δεδομένη χρονική εποχή/ρόλο-χρήστη, τοποθεσία, κτλ. Τέτοιες τεχνικές εκμάθησης μπορούν να αυξήσουν την εμπιστοσύνη στην ακρίβεια της πληροφορίας προφίλ και γενικά μπορούν να βασίζονται σε μετρήσεις και αλληλεπιδράσεις με χρήστες/συσκευές.

Το τμήμα «παρακολούθηση, ανίχνευση και απόκτηση πλαισίου» μπορεί να προσθέσει ευρωστία στις διαδικασίες αποφάσεων και επίσης να παρέχει μέσα για λογική, όχι μόνο για το τι συμβαίνει αυτή τη στιγμή, αλλά και για το τι είναι πιθανό να συμβεί στο μέλλον. Επομένως, οι τεχνικές εκμάθησης μπορούν να αποφέρουν τις τυπικές ικανότητες εναλλακτικών, υποψήφιων διαμορφώσεων, σε ορισμένα πλαίσια. Τέλος, είναι αναγκαίες προβλέψεις πλαισίου για το δυναμικό χειρισμό καταστάσεων.

Το τμήμα «μεσιτεία πόρων και υπηρεσιών» μπορεί να αναπτύξει γνώση σχετικά με τις ικανότητες γειτονικών στοιχείων, για την διευκόλυνση αδιάλειπτης φορητότητας και πανταχού παρούσας πρόνοιας.

Τέλος, το τμήμα «διαπραγμάτευση διαμόρφωσης, επιλογή και εκτέλεση» της λειτουργικότητας, χρησιμοποιεί πληροφορία για τυπικές λύσεις σε σχέση με την επιλεχθείσα συμπεριφορά του σημείου πρόσβασης, έτσι ώστε να αυξηθεί η εμπιστοσύνη στην πιο πιθανή συμπεριφορά πολιτικών, στο χειρισμό κάποιων πλαισίων. Αυτή η γνώση θα επιτρέψει γρηγορότερη και πιο αξιόπιστη επιλογή διαμορφώσεων [47].

РАНЕЕЗНАМО ПЕРПАА

8 Ρομποτικά, Γνωσιακά Σημεία Πρόσβασης (Robotic, Cognitive Access Points - RoboCAPs)

8.1 Η ιδέα των RoboCAPs

Οι ασύρματες επικοινωνίες έχουν διεισδύσει σχεδόν σε όλους τους τομείς της ζωής των ανθρώπων. Έχουν γίνει σημαντικές και επηρεάζουν τόσο την προσωπική όσο και την επαγγελματική μας ζωή. Ενώ οι περισσότεροι τύποι ασύρματων δικτύων, όπως τα κυψελωτά δίκτυα των κινητών τηλεπικοινωνιών, έχουν μελετηθεί εξαντλητικά και έχουν βελτιστοποιηθεί, ώστε να διαχειριστούν ποικίλες καταστάσεις, υπάρχουν επίσης κάποια πλαίσια, τα οποία απαιτούν ευφυή διαχείριση, συγκεκριμένα:

- Κινητά hotspots, π.χ. σε πάρκα ή φεστιβάλ όπου μεγάλα πλήθη μετακινούνται από περιοχή σε περιοχή, ανάλογα με την ώρα και τον τόπο των γεγονότων (π.χ. στο φεστιβάλ του Glastonbury όπου πολλαπλές σκηνές ελκύουν πλήθος ποικίλου μεγέθους, ανάλογα με τον καλλιτέχνη).
- Περιοχές που έχασαν τις υποδομές τους, π.χ. μετά από σεισμό, φωτιά, τρομοκρατική επίθεση ή άλλους τύπους γεγονότων που προκαλούν βλάβες.
- Περιοχές με σκληρή μορφολογία που πρέπει να εξερευνηθούν/παρακολουθηθούν/καλυφθούν, όπως σπηλιές ή δάση και/ή μεγάλες κατασκευαστικές/παραγωγικές υποδομές.

Το κίνητρο της χρήσης των RoboCAPs είναι ο ευφυής χειρισμός τέτοιων σημαντικών πλαισίων, συγκεκριμένα των κινητών hotspots, καταστάσεις συμφορών/βλαβών, περιοχές με σκληρή μορφολογία, που αναπαριστούν ένα σημαντικό ποσοστό των περιπτώσεων που πρέπει να διαχειριστούν από μια ασύρματη υποδομή.

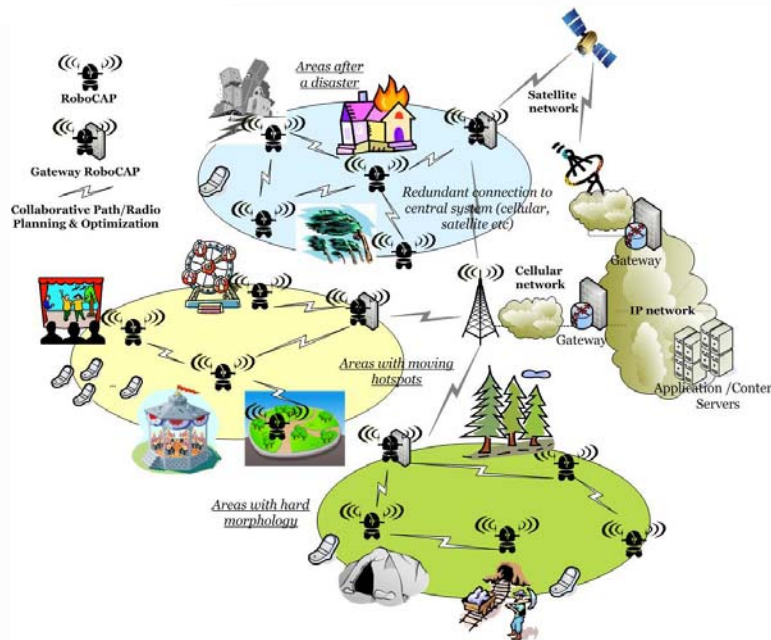
Αυτά τα πλαίσια περικλείουν δύσκολα ασύρματα περιβάλλοντα, στα οποία οι επικοινωνιακές απαιτήσεις (π.χ. κίνηση, φορητότητα) και τα χαρακτηριστικά (π.χ. ράδιο-ποιότητα) μπορούν γρήγορα να αλλάξουν. Σε τέτοια περιβάλλοντα, είναι πρωτεύουσας σημασίας η διαχείριση παροδικών καταστάσεων (αυτές που δεν μπορούν να χαρακτηρισθούν από στατιστική σταθερότητα). Επίσης τέτοια περιβάλλοντα μπορεί να είναι περιοχές γεμάτες εμπόδια, με απρόβλεπτη τοπολογία. Σε τέτοιες ποικίλες περιπτώσεις ο κλασικός εξοπλισμός και ο παραδοσιακός ράδιο-σχεδιασμός μπορεί να μην είναι ευφυής εξαιτίας των συσχετισμένων Πρωτευόντων και Λειτουργικών Εξόδων (CAPEX και OPEX, αντίστοιχα). Ως εκ τούτου, ακόμα απαιτούνται τεχνολογίες για την ευφυή πρόβλεψη των κατάλληλων QoS επιπέδων. Το ευφυές κόστος μπορεί να ζευγαρωθεί με την μείωση των CAPEX και OPEX. Η μείωση των CAPEX μπορεί να

επιτευχθεί με την μη κατασκευή μόνιμων υποδομών. Η μείωση στα OPEX μπορεί να προέλθει από την χρήση γνωσσιακών τεχνικών, αφού σχετίζονται με χαρακτηριστικά αυτό-οργάνωσης. Το QoS αναφέρεται στα επίπεδα της διαθέσιμης υπηρεσίας, των επιδόσεων και της αξιοπιστίας (π.χ. συνέχεια της ράδιο-υπηρεσίας) που παρέχονται στους χρήστες.

Αυτή η εργασία συμβάλλει στην επίτευξη της πρόβλεψης ενός ευφυούς κόστους για το QoS, με την ανάπτυξη και την επικύρωση μιας λύσης που βασίζεται σε Σημεία Πρόσβασης (Access Points - APs) που είναι ικανά να κινούνται αυτόνομα και να εγκαθιστούν ένα ράδιο-δίκτυο σε σύντομο χρονικό διάστημα, με περιορισμένο κεντρικό έλεγχο/διαχείριση. Το ράδιο-δίκτυο προσφέρει ασύρματη πρόσβαση στους χρήστες, και βασίζεται σε ad-hoc συνδεσιμότητα, έχοντας ένα υποσύνολο από στοιχεία που λειτουργούν ως πύλες (gateways) προς υποδομές ευρειών περιοχών (wide-area infrastructure). Η κίνηση μπορεί να βασίζεται σε μια ρομποτική πλατφόρμα (που είναι η κύρια εστίαση αυτής της εργασίας) ή σε ένα αυτοκίνητο (τότε οδηγίες προσφέρονται στον οδηγό) [57]. Συνεπώς, ακολουθώντας αυτή την προσέγγιση, το ευφύες κόστος μπορεί να προκύψει από την έλλειψη ανάγκης για την δημιουργία μόνιμων υποδομών για τη διαχείριση συγκεκριμένων καταστάσεων. Επιπλέον, η αυτόνομη συμπεριφορά των στοιχείων συμβάλλει επίσης στο ευφύες κόστος, αφού αυτό σημαίνει μειωμένα OPEX.

Έχοντας υπόψη τα παραπάνω, η ιδέα αυτής της εργασίας είναι η σχεδίαση, η ανάπτυξη και η επικύρωση Ρομποτικών, Γνωσσιακών Σημείων Πρόσβασης (Robotic, Cognitive Access Points – RoboCAPs), που θα έχουν την ικανότητα να κινούνται αυτόματα και αυτόνομα (βασιζόμενα κυρίως σε ρομποτικές πλατφόρμες, αλλά και σε αυτοκίνητα), να καθορίζουν τις θέσεις τους, να παρέχουν κατάλληλα επίπεδα QoS στους χρήστες, να σχηματίζουν ad-hoc ράδιο-δίκτυα και να διασφαλίζουν σύνδεση με ένα κεντρικό σύστημα υπηρεσιών (ευρείας περιοχής), π.χ. ένα κυβελωτό δίκτυο. Η Εικόνα 8.1 δείχνει την ιδέα των RoboCAPs και κάποια τυπικά σενάρια εφαρμογής τους.

Σκοπός των RoboCAPs δεν είναι να αντικαταστήσουν τα τυπικά APs, π.χ. κυβελωτούς σταθμούς βάσης. Αντίθετα, με το να ολοκληρώνουν τα χαρακτηριστικά και τις εκπληρώσεις των ερευνών τόσο των τηλεπικοινωνιακών δικτύων όσο και των ρομποτικών τομέων, θα είναι ικανά να σχηματίσουν ευέλικτα, γνωσσιακά ad-hoc δίκτυα, που θα συμπληρώνουν τις υπάρχουσες υποδομές και θα προσφέρουν ευφυή πρόβλεψη του κόστους των σχετικών QoS επιπέδων στα απαιτητικά σενάρια χρήσης που αναφέρθηκαν προηγουμένως.



Εικόνα 8.1 Αρχική αρχιτεκτονική των RoboCAPs και περιοχές εφαρμογής.

8.2 Στόχος των RoboCAPs

Ο στόχος είναι η επίλυση ενός σύνθετου προβλήματος βελτιστοποίησης που μπορεί να εκφραστεί ως εξής:

«Δεδομένη πληροφορία και γνώση πάνω:

- Στο πλαίσιο που πρέπει να χειριστεί σε όρους (i) απαίτησης κίνησης, (ii) φορητότητα χρήστη, (iii) ράδιο-ποιότητας, (iv) επιλογών για σύνδεση σε δίκτυα ευρείας περιοχής, (v) των τρεχουσών τοποθεσιών των στοιχείων των RoboCAPs, και των χαρακτηριστικών των ενδεχόμενων διαδρομών που αυτά τα στοιχεία μπορούν να ακολουθήσουν,
- Στις ικανότητες των RoboCAP οντοτήτων σε όρους (i) τηλεπικοινωνιακής δικτύωσης (π.χ. Τεχνολογίες Ράδιο-Πρόσβασης και φάσμα που μπορούν να λειτουργήσουν, κ.α.) και (ii) φυσικής κίνησης (π.χ. πιθανή ταχύτητα του στοιχείου, τύποι διαδρομών και εμπόδια που μπορεί να προσπεράσει, κ.α.),
- Στις ενδεχόμενες πολιτικές που πρέπει να ακολουθηθούν από τις RoboCAP οντότητες και το δίκτυό τους,

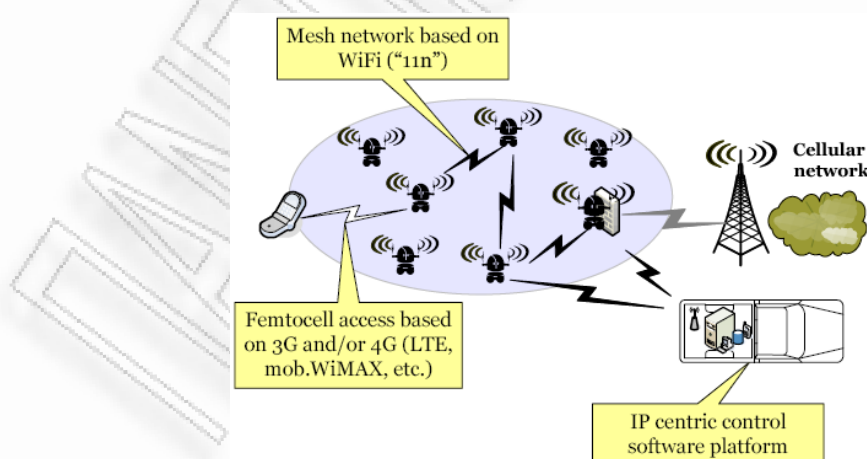
Εύρεση:

- Της βέλτιστης ρύθμισης του ράδιο-δικτύου των RoboCAP οντοτήτων,
- Των βέλτιστων διαδρομών που πρέπει να ακολουθήσουν οι RoboCAP οντότητες,

έτσι ώστε να προσφέρουν επαρκή επίπεδα QoS, σε όρους διαθεσιμότητας της υπηρεσίας, επιδόσεων και αξιοπιστίας».

Η βελτιστοποίηση της ρύθμισης του ράδιο-δικτύου μπορεί να συνδεθεί με παράγοντες όπως οι ζώνες του φάσματος που επιλέχθηκαν για να εξυπηρετήσουν τους χρήστες ή για να διασυνδέσουν τα στοιχεία των RoboCAPs, οι αποφάσεις δρομολόγησης, και οι πύλες που επιλέχθηκαν. Σχετικά με αυτό το θέμα, η χρήση ορισμένων ζωνών του φάσματος, διαδρομών στο ad-hoc δίκτυο και πυλών προς δίκτυο ευρείας περιοχής μπορεί να ενθαρρυνθεί ή να αποθαρρυνθεί. Η βελτιστοποίηση των διαδρομών που επιλέχθηκαν μπορεί να σχετίζεται με παράγοντες όπως το μήκος, η κλίση, πιθανά εμπόδια (διάφορες δυσκολίες γενικά), κ.α. Τα επίπεδα του QoS που προσφέρονται από το RoboCAP δίκτυο θα είναι αυστηροί (και σε μερικές περιπτώσεις μη αυστηροί) περιορισμοί στην βελτιστοποίησή.

Η γενική ιδέα της δικτύωσης των RoboCAPs φαίνεται στην Εικόνα 8.2. Το δίκτυο των RoboCAPs προσφέρει στους χρήστες πρόσβαση μικροκυψέλης. Τα RoboCAPs αποτελούν ένα αυτό-προσαρμοστικό δίκτυο πλέγματος με δυναμική επιλογή πύλης. Κάθε RoboCAP οντότητα λειτουργεί σαν ένα AP και σαν ένας ασύρματος δρομολογητής δικτύου πλέγματος, που μπορεί να γίνει η πύλη για το κυψελωτό δίκτυο, εάν βρίσκεται στη βέλτιστη θέση. Σε περιπτώσεις τραχιών επιφανειών, η φυσική κίνηση μπορεί να αντισταθμιστεί από την τοποθέτηση κεραιών. Κατανεμημένες λύσεις θα ερευνηθούν. Ωστόσο, ο επικαλύπτων IP-κεντρικός έλεγχος θα έχει επίσης βασικό ρόλο. Γενικά εγγυάται την σταθερότητα του δικτύου. Στην περίπτωση που η κατανεμημένη προσέγγιση δεν μπορεί να δώσει λύσεις, ο επικαλύπτων IP-κεντρικός έλεγχος θα εγγυηθεί την σωστή λειτουργία του τμήματος του δικτύου.



Εικόνα 8.2 Στιγμιότυπο της ιδέας της δικτύωσης των RoboCAPs (οι RATs που υποδεικνύονται είναι πιθανές λύσεις).

Η ιδέα θα δουλέψει με κάθε Τεχνολογία Ράδιο-Πρόσβασης, μέσα στα όρια της συγκεκριμένης τεχνολογίας. Η υπόδειξη συγκεκριμένων τεχνολογιών στην Εικόνα 8.2, είναι απλά μια από τις πολλές λύσεις. Καθώς τα RoboCAPs θα γίνονται πραγματικότητα, θα υποστηρίζονται περισσότερες RATs όπως κάνουν και οι ασύρματες υποδομές. Σε όρους φάσματος, ο σκοπός είναι να χρησιμοποιηθεί φάσμα με άδεια, χωρίς άδεια ή αχρησιμοποίητο.

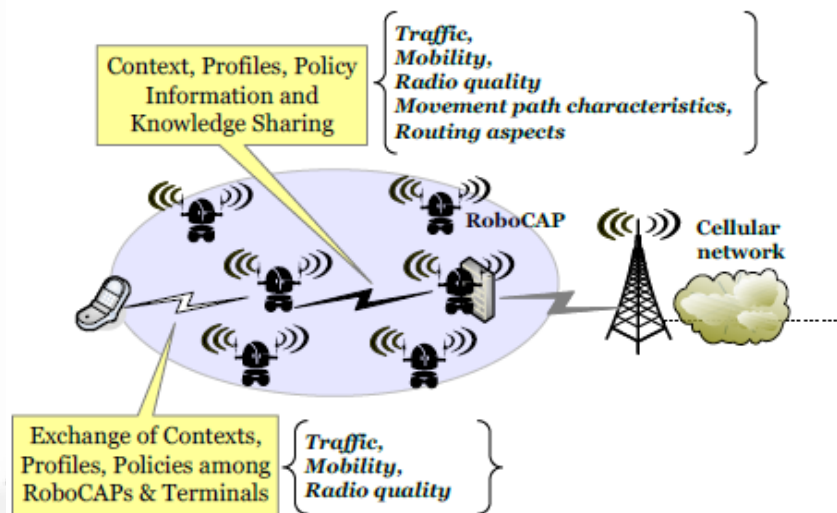
Για να πετύχουν τον κύριο στόχο, τα RoboCAPs θα διεξάγουν εργασίες για την αντιμετώπιση των ακόλουθων τεχνολογικών προκλήσεων:

- Απόκτηση και ανταλλαγή πληροφοριών για το πλαίσιο που αντιμετωπίζουν και το προφίλ και τις πολιτικές που περικλείονται. Οι πληροφορίες του πλαισίου αναφέρονται στις απαιτήσεις της κίνησης, τη φορητότητα του χρήστη, την ράδιο-ποιότητα, τα υποψήφια μονοπάτια και τα χαρακτηριστικά τους, και τις επιλογές σχετικά με την συνδεσιμότητα της ευρείας περιοχής. Η συλλογή των πληροφοριών θα βασίζεται στη συνεργασία των RoboCAPs οντοτήτων και των τερματικών. Τα προφίλ προσφέρουν τις ικανότητες των RoboCAPs οντοτήτων και των τερματικών, και τις απαιτήσεις και τις προτιμήσεις των χρηστών. Οι πολιτικές παρέχουν κανόνες για τον χειρισμό του πλαισίου, σε όρους των στόχων που πρέπει να βελτιστοποιηθούν, των στρατηγικών που θα χρησιμοποιηθούν για τη βελτιστοποίηση, και των περιορισμών που πρέπει να ικανοποιηθούν [53]. Η Εικόνα 8.3 δείχνει την ιδέα της ανταλλαγής των πληροφοριών.
- Ανάπτυξη και αξιολόγηση (μέσω αναλυτικών μοντέλων, προσομοιώσεων και προτυποποιήσεων) των στρατηγικών βελτιστοποίησης του κοινού ράδιο-δικτύου και της επιλογής της διαδρομής, για την ενεργοποίηση της πρόβλεψης ενός ευφυούς κόστους για το QoS ενός αυτό-προσαρμοζόμενου δικτύου πλέγματος σε ένα δυναμικό περιβάλλον. Όπως ήδη ειπώθηκε, η βελτιστοποίηση του ράδιο-δικτύου αναφέρεται στην ρύθμιση της διεπαφής (αέρας), στο σχηματισμό του ad-hoc, αυτό-προσαρμοζόμενου δικτύου πλέγματος των RoboCAPs οντοτήτων και στο δυναμικό προσδιορισμό των πυλών [56]. Η Εικόνα 8.3 δείχνει την λειτουργία των στρατηγικών βελτιστοποίησης, καθώς και το ρόλο όλης της λειτουργικότητας των RoboCAPs. Σχετικά με αυτό το θέμα, η εικόνα μπορεί να φανεί σαν μια περιγραφή υψηλού επιπέδου της λειτουργικής αρχιτεκτονικής των RoboCAPs.
- Ανάπτυξη της γνώσης και του ευφυούς διαμοιρασμού μεταξύ των RoboCAPs οντοτήτων. Η γνώση θα αναπτυχθεί αυτόνομα (π.χ. για κάθε RoboCAP οντότητα) και συλλογικά (π.χ. για ένα σύνολο οντοτήτων). Η γνώση μπορεί να αναφέρεται στα πλαίσια που αντιμετωπίζονται

συνήθως, στις πολιτικές που εφαρμόζονται, στις αποφάσεις βελτιστοποίησης που λήφθηκαν και στην ευφυΐα (σε όρους QoS και κόστους) διάφορων λύσεων και/ή πτυχών που οδηγούν σε λύσεις (π.χ. πολιτικές). Ευφυείς μηχανισμοί διαμοιρασμού, αποσκοπούν στην εύρεση της καλύτερης στρατηγικής για την ανταλλαγή πληροφοριών και γνώσης, λαμβάνοντας υπόψη την ακεραιότητα των δεδομένων και την κατάσταση των RoboCAPs οντοτήτων, σε όρους υπολογιστικών και δικτυακών πόρων. Ουσιαστικά, αυτοί οι μηχανισμοί παρέχουν απαντήσεις στο «τι» να στείλουν και «πότε».

- Προτυποποίηση, επικύρωση και προβολή των RoboCAP δικτύων, που εκτελούν συνεχείς ανανεώσεις των θέσεών τους και των παραμέτρων του ράδιο-δικτύου, ώστε οι χρήστες να καθίστανται με συνδεσιμότητα ευφυούς κόστους κατάλληλου QoS.
- Αντίληψη των δραστηριοτήτων διάδοσης και δημιουργία συνεισφορών προτυποποίησης.

Η Εικόνα 8.3 δείχνει την ανταλλαγή πληροφοριών και γνώσης στα RoboCAPs.

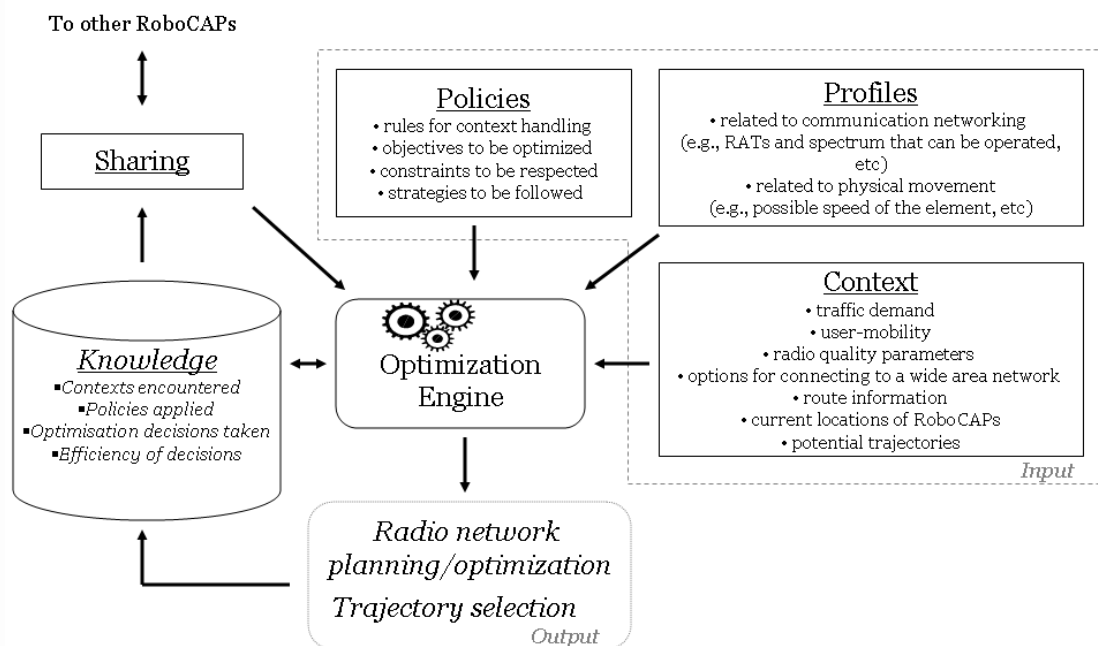


Εικόνα 8.3 Απόκτηση και ανταλλαγή πληροφοριών πλαισίου, προφίλ και πολιτικών, και σχετικής γνώσης στα RoboCAPs.

Η Εικόνα 8.4 δείχνει το ρόλο της λειτουργικότητας που θα αναπτυχθεί από τα RoboCAPs. Δείχνει το ρόλο των στοιχείων που εστιάζουν στο πλαίσιο, στα προφίλ (ικανότητες), στις πολιτικές, στη μηχανή βελτιστοποίησης (εφαρμογή συγκεκριμένων στρατηγικών), στην ανάπτυξη της γνώσης και στο διαμοιρασμό. Το πλαίσιο, τα προφίλ και οι πολιτικές, όπως και η υπάρχουσα γνώση, είναι είσοδος για την μηχανή βελτιστοποίησης. Η μηχανή βελτιστοποίησης παράγει μια συγκεκριμένη έξοδο που εφαρμόζεται για το σχηματισμό του RoboCAPs ράδιο-δικτύου και για την επιλογή των διαδρομών. Η πληροφορία συλλέγεται, ανάλογα με την ευφυΐα των λύσεων, έτσι ώστε να αποκτηθεί καινούρια γνώση. Η λύση του

προβλήματος θα βασίζεται σε τεχνικές βελτιστοποίησης, επαυξημένη με μάθηση μηχανής, ώστε να κερδίσει από την εμπειρία των πλαισίων που αντιμετωπίστηκαν, τις αντίστοιχες λύσεις που επιλέχθηκαν και την ευφυΐα τους (σε όρους κόστους και QoS).

Η Εικόνα 8.5 δείχνει ένα παράδειγμα της λειτουργικότητας των RoboCAPs. Το παράδειγμα περιλαμβάνει ένα hotspot που πρέπει να χειριστεί, τέσσερις RoboCAP οντότητες και δύο ασύρματα τμήματα δικτύου που μπορούν να χρησιμοποιηθούν για συνδεσιμότητα ευρείας περιοχής. Οι τρεις διαδρομές έχουν διαφορετικά χαρακτηριστικά, και ως εκ τούτου, διαφορετικό βαθμό δυσκολίας. Τα δύο ασύρματα τμήματα δικτύου έχουν διαφορετικές ικανότητες QoS.

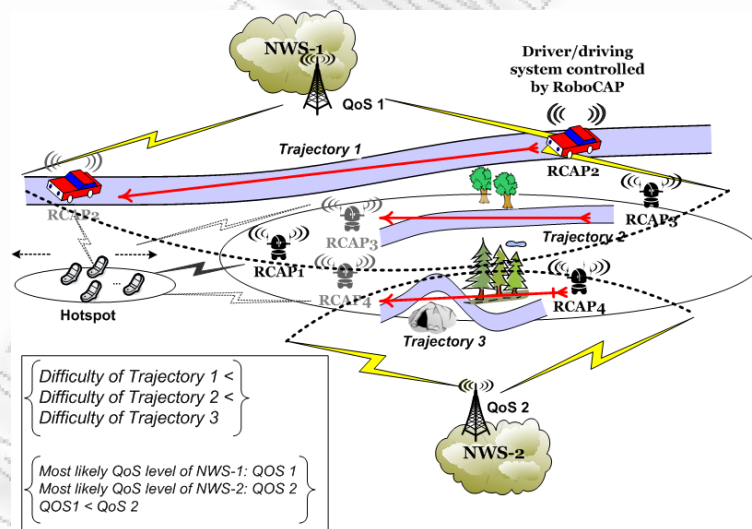


Εικόνα 8.4 Υψηλού επιπέδου λειτουργική αρχιτεκτονική για από κοινού βελτιστοποίηση του ράδιο-δικτύου και της κίνησης των RoboCAP οντοτήτων, ώστε να γίνει χειρισμός του πλαισίου που αντιμετωπίζεται, βασισμένο επίσης σε ικανότητες (προφίλ) και πολιτικές.

Στο παράδειγμα, η RoboCAP οντότητα, που συμβολίζεται με RCAP1, πρέπει να αντιμετωπίσει ένα συγκεκριμένο πλαίσιο που περιλαμβάνει ένα hotspot. Το RCAP1 μπορεί να συνεργαστεί με τρεις επιπλέον οντότητες, που συμβολίζονται ως RCAP2, RCAP3 και RCAP4. Το RCAP2 βρίσκεται πάνω σε αυτοκίνητο, ενώ τα RCAP3, RCAP4 είναι πάνω σε ρομπότ. Για να φτάσουν μια περιοχή από την οποία μπορούν να συμβάλουν στην εξυπηρέτηση του hotspot, το RCAP2 μπορεί να ακολουθήσει μια ευκολότερη διαδρομή σε σύγκριση με το RCAP3, και το RCAP3 μπορεί να ακολουθήσει ένα ευκολότερο μονοπάτι σε σχέση με το RCAP4. Τα RCAP2 και RCAP3 μπορούν να αποκτήσουν και να προσφέρουν συνδεσιμότητα μέσω του τμήματος-δικτύου 1 (NWS1), ενώ το RCAP4 μπορεί να αποκτήσει συνδεσιμότητα ευρείας

περιοχής μέσω του τμήματος-δικτύου 2 (NWS2). Οι RoboCAP οντότητες έχουν γνώση του πιθανού QoS που τα δύο τμήματα δικτύου μπορούν να προσφέρουν. Για παράδειγμα, μπορούμε να υποθέσουμε ότι τα επίπεδα του QoS του NWS1 είναι χαμηλότερα από τα πιο πιθανά επίπεδα του QoS του NWS2. Όπως φαίνεται, μπορεί να υπάρξουν οντότητες που έχουν γνώση του ίδιου πράγματος (π.χ. τα RCAP2 και RCAP3 στο NWS1) ή διαφορετικών πραγμάτων (π.χ. αυτό του RCAP4 στο NWS2).

Εναπόκειται στο RCAP1 (απόφαση μέσω της ευφυΐας του ή μέσω του επικαλύπτοντος IP-κεντρικού ελέγχου) να αποφασίσει πώς θα εξυπηρετήσει καλύτερα το hotspot. Για παράδειγμα, μπορεί να επιλεγθεί το RCAP1 (και το RCAP2), που σημαίνει ότι η βέλτιστη λύση που απαιτείται, είναι η εγκατάσταση ενός γρήγορου, λιγότερο καλού ράδιο-δικτύου (χρήση του NWS1, που προσφέρει χαμηλότερο QoS, μέσω του RCAP1, που μπορεί εύκολα να αναπτυχθεί αφού ακολουθεί ένα εύκολο μονοπάτι). Μια άλλη λύση είναι η επιλογή του RCAP4, που σημαίνει ότι η βέλτιστη λύση είναι η χρήση του καλύτερου πιθανού ράδιο-δικτύου (χρήση του NWS2, που προσφέρει υψηλότερο QoS, μέσω του RCAP2, το οποίο είναι πιο δύσκολο να αναπτυχθεί αφού ακολουθεί ένα πιο τραχύ φυσικό μονοπάτι) [54][55].



Εικόνα 8.5 Παράδειγμα λειτουργίας των RoboCAPs.

Συμπερασματικά, για να αντιμετωπιστεί η πολυπλοκότητα αυτού του προβλήματος και να πετύχουμε τους κύριους στόχους, το σχέδιο των RoboCAPs θα ακολουθήσει μια ολοκληρωτική τεχνική προσέγγιση που θα βασίζεται στην αλληλεπίδραση της βελτιστοποίησης του ράδιο-δικτύου με την αξιολόγηση του βέλτιστου μονοπατιού. Ιδιαίτερη έμφαση δε, όπως θα εξηγηθεί αργότερα, θα δοθεί στην εύρεση των θέσεων των RoboCAPs που παρέχουν το ελάχιστο δυνατό κόστος.

8.3 Μαθηματικό μοντέλο

Σκοπός του προβλήματος είναι η δημιουργία ενός ράδιο-δικτύου που θα μπορεί να εξυπηρετεί τις ανάγκες των χρηστών (δηλαδή το φορτίο που παράγουν όλοι οι χρήστες συνολικά). Οι RoboCAP οντότητες, οι οποίες βρίσκονται σε κάποιες αρχικές θέσεις, θα πρέπει να επιλέξουν κατάλληλες νέες θέσεις ακολουθώντας τη βέλτιστη διαδρομή⁵, έτσι ώστε να δημιουργήσουν ένα ράδιο-δίκτυο.

Έστω ένας γράφος $G(V,E)$ όπου V το σύνολο των κορυφών (κόμβων) και E το σύνολο των πλευρών του γράφου. Έστω U το σύνολο των χρηστών⁶, RC το σύνολο των RoboCAP οντοτήτων και Θ το σύνολο των θέσεων και των υποψήφιων θέσεων των RoboCAP οντοτήτων, των χρηστών και των πυλών. Συμβολίζουμε με $C\Theta_r \subseteq \Theta$ το σύνολο των τρεχουσών θέσεων των RoboCAP οντοτήτων, με $C\Theta_u \subseteq \Theta$ το σύνολο των τρεχουσών θέσεων των χρηστών και με $\Theta_{GW} \subseteq \Theta$ το σύνολο των θέσεων των πυλών. Θεωρούμε επίσης τη μεταβλητή απόφασης X_{ur} για την ανάθεση των χρηστών στα κατάλληλα RoboCAPs, τη μεταβλητή $Y_{r\theta}$ για την αντιστοίχιση των RoboCAPs στις υποψήφιες θέσεις του συνόλου Θ , τη μεταβλητή $W_{u\theta}$ για την αντιστοίχιση των χρηστών σε θέσεις και τη μεταβλητή $Z_{r_1 r_2}$ για να δείξουμε αν δύο RoboCAP οντότητες r_1, r_2 είναι συνδεδεμένες.

Για το λόγο αυτό είναι αναγκαίο να θεωρήσουμε δύο κόστη:

- Το κόστος μετακίνησης $m_c(c\theta(r), n\theta(r))$ μιας RoboCAP οντότητας r , όπου $c\theta(r) \in C\Theta_r$, $n\theta(r) \in \Theta$ είναι η τρέχουσα θέση και η θέση προς την οποία θα μετακινηθεί το r αντίστοιχα. Το κόστος αυτό δίνεται από το χρήστη και είναι συνδεδεμένο με την αντίστοιχη πλευρά του γράφου.
- Το τηλεπικοινωνιακό κόστος το οποίο αποτελείται από δύο μέρη:
 1. το τηλεπικοινωνιακό κόστος $t_c(\vartheta(u), \vartheta(r))$ της σύδεσης των χρηστών με τη RoboCAP οντότητα r , όπου $\vartheta(u)$ είναι η τρέχουσα θέση των χρηστών και $\vartheta(r)$ η επιλεγθείσα θέση του r ,

⁵ Με τον όρο βέλτιστη διαδρομή, δεν αναφερόμαστε πάντα στη συντομότερη, αφού λαμβάνουμε υπόψη και την μορφολογία του εδάφους. Υπάρχει περίπτωση για μια συντομότερη διαδρομή να χρειαστεί περισσότερος χρόνος για τις RoboCAP οντότητες να τη διασχίσουν λόγω εμποδίων, σε σχέση με μια πιο μακρινή διαδρομή η οποία δεν έχει εμπόδια.

⁶ Με τον όρο «χρήστης» δεν αναφερόμαστε απαραίτητα σε έναν άνθρωπο, αλλά μπορεί να είναι μια ομάδα ανθρώπων που βρίσκονται στην ίδια κορυφή του γράφου.

2. το τηλεπικοινωνιακό κόστος $t_c(\vartheta(r_1), \vartheta(r_2))$ της σύνδεσης δύο RoboCAP οντοτήτων r_1 και r_2 , όπου $\vartheta(r_1), \vartheta(r_2)$ είναι οι επιλεχθείσες θέσεις των r_1 και r_2 αντίστοιχα.

Σύμφωνα με τα παραπάνω, η αντικειμενική συνάρτηση που θα πρέπει να ελαχιστοποιηθεί είναι η εξής:

$$f(u, r) = \left(\sum_u \sum_r (X_{ur} \cdot t_c(\theta(u), \theta(r))) \right) + \left(\sum_r \sum_\theta (Y_{r\theta} \cdot m_c(c\theta(r), n\theta(r))) \right) + \left(\sum_{r_1, r_2} (Z_{r_1 r_2} \cdot t_c(\theta(r_1), \theta(r_2))) \right) \quad (8.1)$$

Τα στοιχεία της πρώτης παρένθεσης αποτελούν το τηλεπικοινωνιακό κόστος της σύνδεσης όλων των χρηστών με όλες τις RoboCAP οντότητες. Αντίστοιχα τα στοιχεία της δεύτερης παρένθεσης αποτελούν το συνολικό κόστος της μετακίνησης των RoboCAPs από τις αρχικές τους θέσεις στις επιλεχθείσες, και τα στοιχεία της τρίτης παρένθεσης αποτελούν το συνολικό τηλεπικοινωνιακό κόστος της σύνδεσης μεταξύ των RoboCAPs.

Η αντικειμενική συνάρτηση θα πρέπει να ικανοποιεί τους εξής περιορισμούς:

- Η θέση μιας RoboCAP οντότητας r , δίνεται από τη σχέση

$$\theta(r) = \sum_\theta (Y_{r\theta} \cdot \theta), \quad \forall r \in RC \quad (8.2)$$

- Η θέση ενός χρήστη δίνεται από τη σχέση

$$\theta(u) = \sum_\theta (W_{u\theta} \cdot \theta), \quad \forall u \in U \quad (8.3)$$

- Κάθε χρήστης μπορεί να βρίσκεται σε μία θέση μόνο, οπότε

$$\sum_\theta W_{u\theta} = 1, \quad \forall u \in U \quad (8.4)$$

$$\sum_u W_{u\theta} \leq 1, \quad \forall \theta \in \Theta \quad (8.5)$$

- Κάθε χρήστης εξυπηρετείται από ένα και μόνο RoboCAP, οπότε

$$\sum_r X_{ur} = 1, \quad \forall u \in U \quad (8.6)$$

- Το φορτίο L μιας RoboCAP οντότητας r_1 ισούται με το άθροισμα του φορτίου όλων των χρηστών που εξυπηρετούνται από το r_1 συν το άθροισμα του φορτίου όλων των RoboCAPs με τα οποία το r_1 είναι συνδεδεμένο:

$$L(r_1) = \sum_u (X_{ur_1} \cdot L(u)) + \sum_r (Z_{rr_1} \cdot L(r)) \quad (8.7)$$

- Η χωρητικότητα $cp(r_1)$ ενός RoboCAP r_1 , πρέπει να είναι τουλάχιστον ίση με το άθροισμα του φορτίου όλων των RoboCAPs με τα οποία είναι συνδεδεμένο συν το άθροισμα του φορτίου όλων των χρηστών που εξυπηρετούνται από το r_1 :

$$\sum_u (X_{ur_1} \cdot L(u)) + \sum_r (Z_{rr_1} \cdot L(r)) \leq cp(r_1) \quad (8.8)$$

РАНЕЕЗНАМО ПЕРПАА

9 Προσομοίωση λειτουργίας των RoboCAPs

9.1 Εισαγωγή στο πρόβλημα των RoboCAPs

Σκοπός της εργασίας είναι η μελέτη της λειτουργίας των RoboCAPs. Το πρόβλημα το οποίο μελετάται είναι η εύρεση της βέλτιστης θέσης των RoboCAPs έτσι ώστε να έχουμε ελάχιστο τηλεπικοινωνιακό και μεταφορικό κόστος.

Θεωρούμε ένα γράφο, στις κορυφές του οποίου βρίσκονται όλες οι οντότητες του προβλήματος, δηλαδή οι Σταθμοί Βάσης, τα RoboCAPs και οι χρήστες⁷. Οι κορυφές έχουν συντεταγμένες, ενώ σε κάθε πλευρά του γράφου που ενώνει μια κορυφή i με μια κορυφή j , αντιστοιχίζεται ένας αριθμός d_{ij} , ο οποίος ισούται με το κόστος μετακίνησης από την κορυφή i στην κορυφή j . Για κάθε RoboCAP οντότητα που μετακινείται από μια κορυφή i σε μια κορυφή j , το κόστος μετακίνησης m_c αυξάνεται κατά d_{ij} . Αν δύο οντότητες είναι συνδεδεμένες (π.χ. δύο RoboCAPs, ένα RoboCAP κι ένας Σταθμός Βάσης, ένα RoboCAP κι ένας χρήστης), τότε το τηλεπικοινωνιακό κόστος αυτής της σύνδεσης δίνεται από την παρακάτω σχέση που προκύπτει από το Free Space Loss:

$$t_c = \left(\frac{4\pi d_{ij} f}{c} \right)^2 \quad (9.1)$$

όπου d_{ij} η Ευκλείδεια απόσταση των δύο οντοτήτων, f η συχνότητα των RoboCAPs και c η ταχύτητα του φωτός. Το συνολικό κόστος μιας λύσης (δηλαδή μιας ανάθεσης θέσεων στα RoboCAPs) ίσουςται με το άθροισμα των κοστών μετακίνησης και των τηλεπικοινωνιακών κοστών.

Κάθε Σταθμός Βάσης υπολογίζει το τηλεπικοινωνιακό κόστος με κάθε RoboCAP κι εκείνο που έχει το μικρότερο επιλέγεται για να λειτουργήσει ως πύλη προς τον αντίστοιχο Σταθμό. Επίσης κάθε RoboCAP υπολογίζει το τηλεπικοινωνιακό κόστος σύνδεσης με κάθε χρήστη, κι εκείνο που έχει το μικρότερο θα εξυπηρετήσει τον αντίστοιχο χρήστη. Τέλος, αν ένα RoboCAP βρεθεί μέσα στην ακτίνα κάλυψης ενός άλλου, συνδέονται μεταξύ τους και το φορτίο του ενός μεταφέρεται στο άλλο.

9.2 Περιγραφή των αλγορίθμων

Για την επίλυση του προβλήματος αναπτύχθηκαν δύο αλγόριθμοι. Ο ένας είναι με χρήση του Simulated Annealing, ενώ ο δεύτερος είναι μια παραλλαγή του

⁷ Για λόγους απλότητας, με τον όρο «χρήστης» δεν αναφερόμαστε αποκλειστικά σε έναν άνθρωπο, αλλά μπορεί να είναι μια ομάδα ατόμων που βρίσκονται στην ίδια κορυφή του γράφου.

Simulated Annealing, ο οποίος είναι εμπλουτισμένος με στοιχεία από το Ant Colony Optimization.

9.2.1 Αλγόριθμος Simulated Annealing

Ο χρήστης δίνει μια αρχική θέση στα RoboCAPs, οπότε αυτό χρησιμοποιείται ως μια αρχική λύση για την εκκίνηση του Simulated Annealing. Ο αλγόριθμος ξεκινά και δημιουργεί μια γειτονική λύση της αρχικής με τον εξής τρόπο: Επιλέγεται τυχαία ένα RoboCAP και μετακινείται σε μια τυχαία γειτονική θέση. Τότε υπολογίζεται το κόστος μετακίνησης και τα νέα τηλεπικοινωνιακά κόστη που προκύπτουν καθώς και επανακαθορίζονται τα RoboCAPs που θα λειτουργούν ως πύλες, τα RoboCAPs που θα εξυπηρετούν τους χρήστες και ποια RoboCAPs είναι συνδεδεμένα μεταξύ τους. Αν το νέο ολικό κόστος είναι μικρότερο από το προηγούμενο, τότε η νέα λύση θα είναι και η τρέχουσα βέλτιστη λύση. Αν αντίθετα το νέο κόστος είναι μεγαλύτερο από το παλιό, τότε παράγεται ένας τυχαίος αριθμός $\delta \in [0,1]$. Αν $\delta \leq e^{-\frac{\Delta \text{κόστος}}{CT}}$, όπου $\Delta \text{κόστος}$ η διαφορά του νέου κόστους από το παλιό βέλτιστο κόστος και CT η τρέχουσα θερμοκρασία, τότε το τρέχον βέλτιστο κόστος θα ισούται με το νέο κόστος, διαφορετικά παραμένει ως έχει. Με αυτόν τον τρόπο εξασφαλίζεται ότι θα αποφευχθεί κάποιο τοπικό ελάχιστο. Ο αλγόριθμος επαναλαμβάνεται μέχρι η θερμοκρασία να γίνει μικρότερη από 10^{-30} ή μέχρι να ξεπεραστούν οι 100.000 επαναλήψεις. Στο τέλος εξάγει τη θέση των RoboCAPs και το υπό-βέλτιστο (suboptimal) κόστος της αντικειμενικής συνάρτησης.

9.2.2 Αλγόριθμος Simulated Annealing με στοιχεία Ant Colony Optimization

Στα πλαίσια της εργασίας, αναπτύχθηκε ένας νέος, υβριδικός αλγόριθμος που συνδυάζει δύο αλγορίθμους: τον αλγόριθμο Simulated Annealing και το Ant Colony Optimization. Ο λόγος της επιλογής των δύο αλγορίθμων είναι ότι είναι ευρέως γνωστοί και δοκιμασμένοι στον κόσμο των τηλεπικοινωνιών, και χρησιμοποιούνται ακόμα και για την επίλυση των πιο σύγχρονων προβλημάτων.

Γενικά, σε κάθε βήμα του αλγορίθμου SA, η τρέχουσα λύση αντικαθίσταται από μια τυχαία γειτονική λύση, που επιλέγεται με πιθανότητα που εξαρτάται από την διαφορά των τιμών μιας αντικειμενικής συνάρτησης και μιας παραμέτρου T (θερμοκρασία). Με αυτόν τον τρόπο ο αλγόριθμος ερευνά τον χώρο των λύσεων και αποφεύγει τα τοπικά ελάχιστα.

Ο αλγόριθμος ACO είναι εμπνευσμένος από τον τρόπο που τα μυρμήγκια συλλέγουν τροφή. Όσο μεγαλύτερη ποσότητα φερομόνης βρίσκεται σε ένα μονοπάτι, τόσο καλύτερη θεωρείται αυτή η λύση (π.χ. είναι πιο σύντομη) κι έτσι είναι πιο πιθανό να ακολουθήσει αυτό το μονοπάτι κι ένα άλλο μυρμήγκι.

Στον υβριδικό αλγόριθμο που κατασκευάστηκε, μια γειτονική λύση παρασκευάζεται από την τυχαία μετακίνηση ενός RoboCAP από έναν κόμβο i σε έναν κόμβο j του γράφου. Τότε η πιθανότητα με την οποία επιλέγεται μια νέα λύση, εξαρτάται από τον μηχανισμό της φερομόνης του ACO, και πιο συγκεκριμένα από την παραλλαγή Ant System που περιγράφηκε στην αντίστοιχη ενότητα. Τα βήματα του αλγορίθμου είναι τα εξής:

Βήμα 1: Δίνεται μια αρχική λύση του προβλήματος (π.χ. οι αρχικές θέσεις των οντοτήτων του προβλήματος), μία αρχική θερμοκρασία και ένας ρυθμός ψύξης (cooling rate). Επίσης, υπολογίζεται η αρχική τιμή της αντικειμενικής συνάρτησης.

Βήμα 2: Η αρχική λύση εισάγεται στον υβριδικό αλγόριθμο SA-ACO όπως συμβαίνει και στον απλό SA.

Βήμα 3: Υπολογίζεται η πιθανότητα μετακίνησης ενός RoboCAP από τη θέση i σε μια γειτονική θέση j από τη σχέση

$$p_{ij} = \frac{\varphi_{ij}}{\sum_k \varphi_{ik}} \quad (9.2)$$

όπου φ_{ij} η ποσότητα της φερομόνης στην πλευρά (i,j) του γράφου και k οι γειτονικές κορυφές στην i .

Βήμα 4: Μια γειτονική λύση δημιουργείται ως εξής: Ένα τυχαίο RoboCAP μετακινείται από έναν κόμβο i σε έναν κόμβο j του γράφου. Υπολογίζεται η νέα τιμή της αντικειμενικής συνάρτησης και ταυτόχρονα απελευθερώνεται στην πλευρά (i,j) του γράφου μια ποσότητα φερομόνης ίση με το αντίστροφο του μέσου όρου του αρχικού και του τελικού κόστους:

$$\varphi_{ij} = \frac{1}{0,5 \cdot [TC(i) + TC(j)]} = \frac{2}{TC(i) + TC(j)} \quad (9.3)$$

όπου $TC(i)$, $TC(j)$ το αρχικό ολικό κόστος πριν τη μετακίνηση και το τελικό ολικό κόστος μετά τη μετακίνηση αντίστοιχα. Ταυτόχρονα, τοποθετείται και στις υπόλοιπες πλευρές του γράφου μια ποσότητα φερομόνης η οποία δίνεται από τη σχέση:

$$\varphi_{n_1 n_2} = \frac{1}{TC(i) + TC(j)}, \quad \forall n_1 \neq i, n_2 \neq j, n_1, n_2 \in V \quad (9.4)$$

Επίσης επαναυπολογίζονται η τιμή της αντικειμενικής συνάρτησης, οι πύλες προς τους Σταθμούς Βάσης, ποια RoboCAPs εξυπηρετούν τους χρήστες, ποια RoboCAPs είναι συνδεδεμένα, καθώς και ο πίνακας πιθανοτήτων.

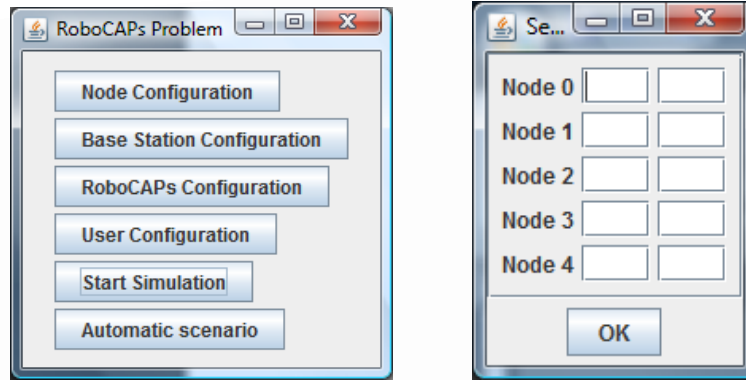
Βήμα 5: Η λύση αξιολογείται όπως και στον απλό SA και η θερμοκρασία μειώνεται κατά τον ρυθμό ψύξης. Αν η θερμοκρασία φτάσει τους 0 βαθμούς διακόπτεται η λειτουργία του αλγορίθμου κι εξάγεται η βέλτιστη λύση.

Για να γίνει επιλογή της γειτονικής κορυφής στην οποία θα μετακινηθεί ένα RoboCAP χρησιμοποιούμε την εξής τεχνική: Θεωρούμε πως η κορυφή i έχει k γειτονικές κορυφές με τις οποίες ενώνεται μέσω των πλευρών $(i,1), (i,2), \dots, (i,k)$ και με πιθανότητες μετάβασης $p_{i1}, p_{i2}, \dots, p_{ik}$. Παράγεται ένας τυχαίος πραγματικός αριθμός $r \in [0,1)$. Αν $r < p_{i1}$ τότε το RoboCAP μεταβαίνει στην κορυφή 1. Αν $\sum_{j=1}^{m-1} p_{ij} \leq r < \sum_{j=1}^m p_{ij}$ τότε μεταβαίνει στην κορυφή m . Τέλος, αν $r \geq \sum_{j=1}^{k-1} p_{ij}$ τότε μεταβαίνει στην κορυφή k .

9.3 Οδηγός χρήσης του προγράμματος εξομοίωσης

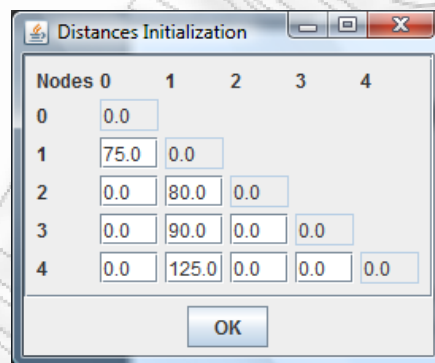
Στα πλαίσια της εργασίας για τη μελέτη της λειτουργίας των RoboCAPs, αναπτύχθηκε πρόγραμμα εξομοίωσης στο οποίο ο χρήστης εισάγει δεδομένα για την τοπολογία του προβλήματος (δηλαδή τα χαρακτηριστικά του γράφου πάνω στον οποίο βρίσκονται οι RoboCAP οντότητες, οι χρήστες και οι Σταθμοί Βάσης), για τα χαρακτηριστικά των RoboCAPs (δηλαδή τον αριθμό των RoboCAPs, την αρχική τους θέση, τη χωρητικότητα της κάθε συσκευής, τη συχνότητα λειτουργίας και την ακτίνα μέσα στην οποία δύο RoboCAPs μπορούν να επικοινωνήσουν μεταξύ τους), για τα χαρακτηριστικά των χρηστών (δηλαδή τον αριθμό τους, τη θέση τους και το φορτίο που φέρουν) και για τα χαρακτηριστικά των Σταθμών Βάσης (δηλαδή τον αριθμό και τη θέση τους). Το πρόγραμμα εξάγει τα κόστη των βέλτιστων λύσεων σε κάθε βήμα του αλγορίθμου, την τελική θέση των RoboCAPs, έναν πίνακα ο οποίος δείχνει ποια RoboCAPs είναι συνδεδεμένα μεταξύ τους, καθώς και ποια είναι τα βέλτιστα RoboCAPs για την εξυπηρέτηση του κάθε χρήστη και για τη σύνδεση με τους Σταθμούς Βάσης.

Στην αρχική οθόνη του προγράμματος ο χρήστης καλείται να επιλέξει τη ρύθμιση που θέλει να κάνει. Αρχικά πρέπει να διαμορφώσει την τοπολογία του προβλήματος πατώντας την επιλογή "Node Configuration" (Εικόνα 9.1(α)).



Εικόνα 9.1 (α) Αρχική οθόνη. (β) Εισαγωγή συντεταγμένων των κόμβων.

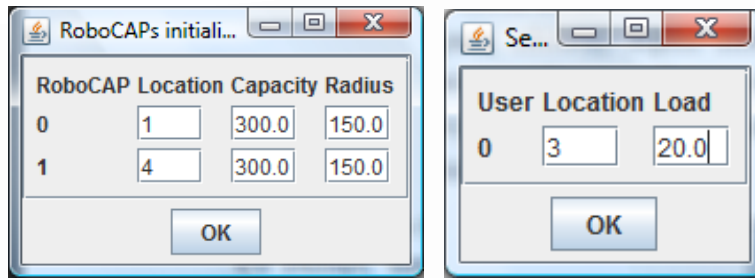
Στη συνέχεια, αφού δηλώσει τον αριθμό των κόμβων του γράφου, εισάγει τις συντεταγμένες του κάθε κόμβου (Εικόνα 9.1(β)). Αφού εισάγει και τις συντεταγμένες, στην επόμενη οθόνη γίνεται εισαγωγή των αποστάσεων των κόμβων οι οποίοι είναι συνδεδεμένοι μεταξύ τους (Εικόνα 9.2). Αν π.χ. ο κόμβος i είναι συνδεδεμένος με τον κόμβο j και η απόστασή τους είναι 150, συμπληρώνουμε στο αντίστοιχο κουτάκι την τιμή 150.0. Αν δύο κόμβοι δεν συνδέονται μεταξύ τους αφήνουμε την τιμή 0.0.



Εικόνα 9.2 Εισαγωγή αποστάσεων συνδεδεμένων κόμβων.

Στη συνέχεια πρέπει να γίνει η ρύθμιση των Σταθμών Βάσης από την αρχική οθόνη με την επιλογή "Base Station Configuration". Ο χρήστης δηλώνει τον αριθμό των Σταθμών Βάσης που υπάρχουν στο πρόβλημα και τον κόμβο στον οποίο βρίσκεται ο καθένας.

Έπειτα, γίνεται η ρύθμιση των RoboCAPs με την επιλογή "RoboCAPs Configuration" της αρχικής οθόνης. Αφού δοθεί ο αριθμός των RoboCAPs και η συχνότητα στην οποία επικοινωνούν, ο χρήστης εισάγει τον κόμβο στον οποίο βρίσκεται κάθε RoboCAP, τη χωρητικότητα που έχει καθώς και την ακτίνα μέσα στην οποία μπορεί να επικοινωνήσει με άλλα RoboCAPs (Εικόνα 9.3(α)).



Εικόνα 9.3 (α) Ρύθμιση των RoboCAPs. (β) Ρύθμιση των χρηστών.

Τέλος, γίνεται η ρύθμιση των χρηστών με την επιλογή “User Configuration” της αρχικής οθόνης. Ο χρήστης εισάγει τον αριθμό των χρηστών, τον κόμβο στον οποίο βρίσκονται και το φορτίο που φέρουν (Εικόνα 9.3(β)).

Με το κουμπί Automatic scenario, επιλέγεται η εκτέλεση ενός αυτοματοποιημένου σεναρίου χρήσης.

Αφού τελειώσει η εισαγωγή των δεδομένων, ο αλγόριθμος ξεκινά να υπολογίζει τα κόστη και τις θέσεις των RoboCAPs με την επιλογή “Start Simulation”. Αφού τελειώσει όλη η διαδικασία, δημιουργούνται τα εξής αρχεία:

- NScost_simple.txt, NScost_aco.txt, NS_ex, NS_auto: Εδώ τυπώνεται το τρέχον κόστος κάθε 50 επαναλήψεις για τον απλό αλγόριθμο, τον υβριδικό, την εξαντλητική αναζήτηση και την εξαντλητική αναζήτηση του αυτοματοποιημένου σεναρίου αντίστοιχα.
- RConnect_simple.txt, RConnect_aco.txt, RConnect_ex, RConnect_auto: Εδώ τυπώνεται ο πίνακας που δείχνει ποια RoboCAPs είναι συνδεδεμένα μεταξύ τους.
- Statistics_simple.txt, Statistics_aco.txt, Statistics_ex, Statistics_auto: Εδώ τυπώνονται για κάθε Σταθμό Βάσης το καλύτερο RoboCAP το οποίο λειτουργεί ως πύλη (gateway) προς το σταθμό αυτό, τα RoboCAPs που εξυπηρετούν κάθε χρήστη, και τέλος τις τελικές θέσεις και τις χωρητικότητες όλων των RoboCAPs.

9.4 Παραδείγματα

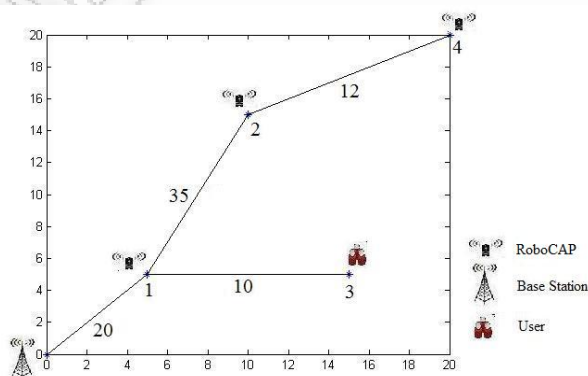
Τα παρακάτω παραδείγματα εκτελέστηκαν με χρήση του προγράμματος εξομοίωσης που περιγράφηκε στην προηγούμενη ενότητα. Το σύστημα της εξομοίωσης αποτελείται από επεξεργαστή Intel Core 2 Duo T8100 (2,10 GHz), 4 GB RAM και περιβάλλον Microsoft Windows Vista SP2.

Τα αποτελέσματα των παραδειγμάτων δίνονται σε Πίνακες και Εικόνες. Οι Εικόνες με τους γράφους των προβλημάτων δείχνουν την αρχική λύση του κάθε προβλήματος. Οι αριθμοί στις πλευρές των γράφων δηλώνουν το κόστος

μετακίνησης, ενώ οι αριθμοί στους κόμβους είναι ο αύξων αριθμός του κόμβου. Οι Πίνακες με τις λεπτομέρειες των Σταθμών Βάσης δείχνουν τα χαρακτηριστικά των Σταθμών Βάσης. Η 1η στήλη αντιστοιχεί στους Σταθμούς Βάσης, η 2η στην κορυφή του γράφου στην οποία βρίσκεται ο κάθε σταθμός και η 3η δείχνει ποιο RoboCAP λειτουργεί ως πύλη προς κάθε Σταθμό. Οι Πίνακες με τις λεπτομέρειες των χρηστών λειτουργούν ως εξής: Η 1η στήλη αντιστοιχεί στους χρήστες, η 2η δείχνει τη θέση των χρηστών στις κορυφές του γράφου, η 3η δείχνει το RoboCAP που εξυπηρετεί κάθε χρήστη και η 4η δείχνει το φορτίο που φέρει κάθε χρήστη. Οι Πίνακες με τις λεπτομέρειες των RoboCAPs είναι και οι πιο σημαντικοί αφού παρουσιάζουν τη βέλτιστη λύση για κάθε αλγόριθμο. Συγκεκριμένα, η 1η στήλη αντιστοιχεί στις RoboCAP οντότητες, η 2η στην κορυφή του γράφου στην οποία βρίσκεται κάθε RoboCAP κατά τη βέλτιστη λύση και η 3η στην χωρητικότητα του κάθε RoboCAP. Οι Πίνακες των συνδεδεμένων RoboCAPs δείχνουν ποιά RoboCAPs είναι συνδεδεμένα. Αν το στοιχείο $[i,j]$ του πίνακα είναι 0 σημαίνει πως το RoboCAP i δεν είναι συνδεδεμένο με το RoboCAP j . Αν είναι 1 τότε είναι συνδεδεμένα. Οι Πίνακες φερομόνης λειτουργούν ως εξής: Το στοιχείο $[i,j]$ του πίνακα ισούται με την ποσότητα της φερομόνης που έχει τοποθετηθεί πάνω στην πλευρά του γράφου που ενώνει την κορυφή i με την κορυφή j . Οι Πίνακες πιθανοτήτων είναι ως εξής: Το στοιχείο $[i,j]$ του πίνακα ισούται με την πιθανότητα που υπάρχει για ένα RoboCAP να μεταβεί από την κορυφή i στην κορυφή j . Τέλος, οι Εικόνες Κόστους – Αριθμού Επανάληψεων παρουσιάζουν τη διακύμανση των τιμών του κόστους για κάθε επανάληψη των αλγορίθμων.

9.4.1 Παράδειγμα 1

Δίνεται ο γράφος όπως φαίνεται στην Εικόνα 9.4. Θεωρούμε ένα Σταθμό Βάσης στον κόμβο 0, τρία RoboCAPs στους κόμβους 1,2 και 4 αντίστοιχα με χωρητικότητα 150 Erlangs και ακτίνα 35m και ένα χρήστη στον κόμβο 3 με φορτίο 20 Erlangs. Τα RoboCAPs επικοινωνούν μεταξύ τους και με τους χρήστες μέσω του πρωτοκόλλου 802.11g [58]. Αναζητείται εκείνη η ανάθεση θέσεων στα RoboCAPs που να προσφέρει το βέλτιστο δυνατό κόστος.



Εικόνα 9.4 Γράφος του Προβλήματος 1.

9.4.1.1 Λύση μέσω εξαντλητικής αναζήτησης

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες. Ο χρόνος εκτέλεσης ήταν 235 ms.

Πίνακας 9.1 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.

BS	Location	RC Gateway
0	0	0

Πίνακας 9.2 Λεπτομέρειες χρηστών του Προβλήματος 1.

User	Location	Connected RC	Load
0	3	2	20.0

Πίνακας 9.3 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.

RC	Location	Capacity
0	2	150.0
1	4	150.0
2	3	150.0

Πίνακας 9.4 Συνδεδεμένα RoboCAPs του Προβλήματος 1.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

9.4.1.2 Λύση με χρήση του αλγορίθμου *Simulated Annealing*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 4806 ms.

Πίνακας 9.5 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.

BS	Location	RC Gateway
0	0	0

Πίνακας 9.6 Λεπτομέρειες χρηστών του Προβλήματος 1.

User	Location	Connected RC	Load
0	3	0	20.0

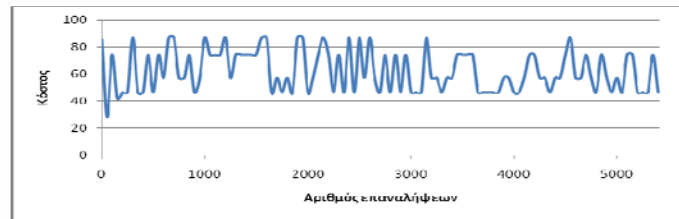
Πίνακας 9.7 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.

RC	Location	Capacity
0	3	150.0
1	1	150.0
2	2	150.0

Πίνακας 9.8 Συνδεδεμένα RoboCAPs του Προβλήματος 1.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

Παρατηρούμε πως επειδή τα RoboCAPs βρίσκονται σε κοντινή απόσταση και το ένα βρίσκεται μέσα στην ακτίνα κάλυψης του άλλου, και τα τρία RoboCAPs είναι συνδεδεμένα μεταξύ τους.



Εικόνα 9.5 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 1.

9.4.1.3 Λύση με χρήση του αλγορίθμου *Simulated Annealing* με στοιχεία ACO

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 4680 ms.

Πίνακας 9.9 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 1.

BS	Location	RC Gateway
0	0	1

Πίνακας 9.10 Λεπτομέρειες χρηστών του Προβλήματος 1.

User	Location	Connected RC	Load
0	3	0	20.0

Πίνακας 9.11 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 1.

RC	Location	Capacity
0	1	150.0
1	0	150.0
2	2	150.0

Πίνακας 9.12 Συνδεδεμένα RoboCAPs του Προβλήματος 1.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

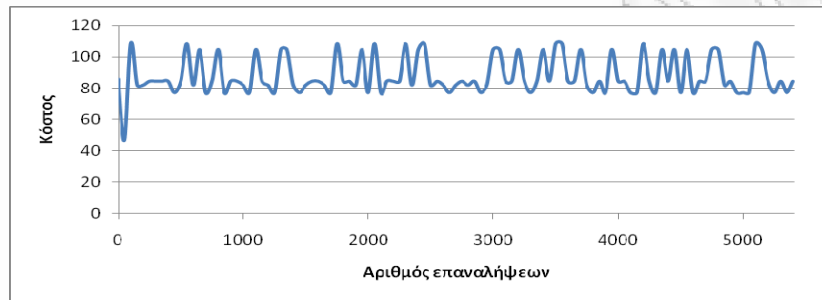
Πίνακας 9.13 Πίνακας φερομόνης του Προβλήματος 1.

Nodes	0	1	2	3	4
0	0	62,38777	0	0	0
1	62,38777	0	50,81469	39,83565	0
2	0	50,81469	0	0	39,84102
3	0	39,83565	0	0	0
4	0	0	39,84102	0	0

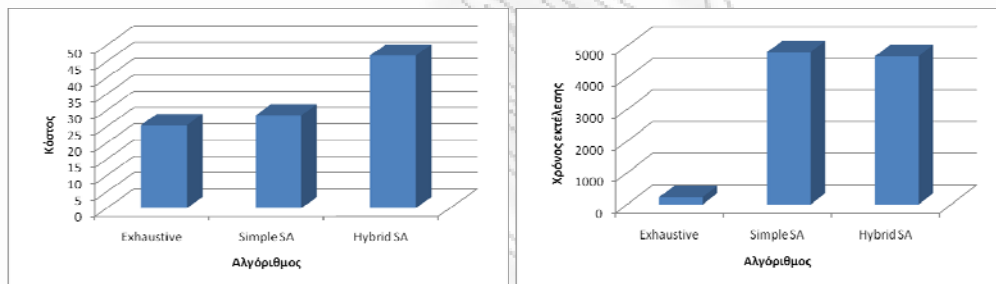
Πίνακας 9.14 Πίνακας πιθανοτήτων του Προβλήματος 1.

Nodes	0	1	2	3	4
0	0	1	0	0	0
1	0,407662	0	0,332039	0,260299	0
2	0	0,560524	0	0	0,439476
3	0	1	0	0	0
4	0	0	1	0	0

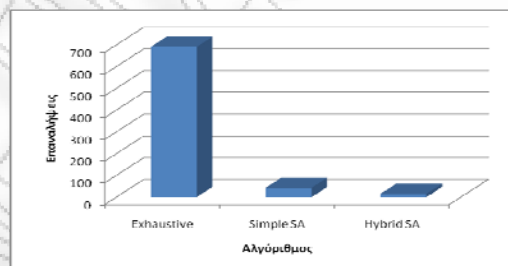
Παρατηρούμε πως όπως αναμενόταν, το άθροισμα των στοιχείων κάθε γραμμής του πίνακα πιθανοτήτων ισούται με 1, εφόσον πρόκειται για πιθανότητες.



Εικόνα 9.6 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 1.



Εικόνα 9.7 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.



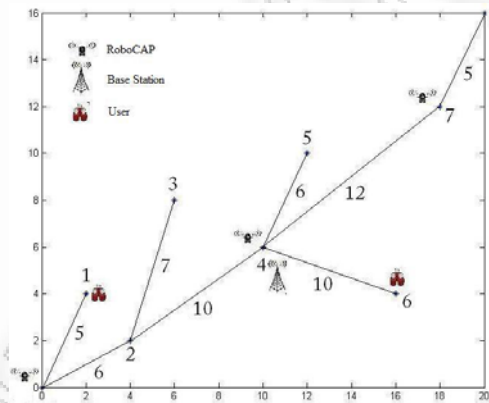
Εικόνα 9.8 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.

Από τα παραπάνω γραφήματα (Εικόνα 9.7(α)) παρατηρούμε πως το βέλτιστο κόστος του απλού αλγορίθμου Simulated Annealing είναι πιο κοντά στην πραγματική λύση που υπολογίζεται μέσω της εξαντλητικής αναζήτησης. Παρ' όλα αυτά, ο υβριδικός αλγόριθμος SA-ACO χρειάζεται λιγότερο χρόνο για να εκτελεστεί (Εικόνα 9.7(β)). Επίσης, ο υβριδικός αλγόριθμος χρειάζεται μικρότερο αριθμό επαναλήψεων μέχρι να υπολογίσει το βέλτιστο κόστος του (Εικόνα 9.8). Τέλος,

πρέπει να παρατηρηθεί πως λόγω του μικρού μεγέθους του προβλήματος (δηλαδή του αριθμού των κόμβων, του αριθμού των RoboCAPs, του αριθμού των χρηστών και του αριθμού των Σταθμών Βάσης) η εξαντλητική αναζήτηση ολοκληρώνεται σε πολύ μικρότερο χρόνο σε σχέση με τους άλλους δύο αλγορίθμους, όμως χρειάζεται πολύ μεγαλύτερο αριθμό επαναλήψεων μέχρι να υπολογίσει το πραγματικό ελάχιστο κόστος.

9.4.2 Παράδειγμα 2

Δίνεται ο γράφος όπως φαίνεται στην Εικόνα 9.9. Θεωρούμε ένα Σταθμό Βάσης στην κορυφή 4 του γράφου, τρία RoboCAPs στις κορυφές 0, 4 και 7 αντίστοιχα και δύο χρήστες στις κορυφές 1 και 6. Κάθε RoboCAP έχει χωρητικότητα 300 Erlangs και ακτίνα κάλυψης 35m, ενώ κάθε χρήστης έχει φορτίο 30 Erlangs. Η επικοινωνία μεταξύ των RoboCAPs και μεταξύ των RoboCAPs και των χρηστών γίνεται μέσω του πρωτοκόλλου 802.11g [58]. Ζητείται να βρεθεί η θέση των RoboCAPs από την οποία θα προκύπτει βέλτιστο κόστος.



Εικόνα 9.9 Γράφος του Προβλήματος 2.

9.4.2.1 Λύση μέσω εξαντλητικής αναζήτησης

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες. Ο χρόνος εκτέλεσης ήταν 1218 ms.

Πίνακας 9.15 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.

BS	Location	RC Gateway
0	4	1

Πίνακας 9.16 Λεπτομέρειες χρηστών του Προβλήματος 2.

User	Location	Connected RC	Load
0	1	2	30.0
1	6	2	30.0

Πίνακας 9.17 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.

RC	Location	Capacity
0	0	300.0
1	2	300.0
2	1	300.0

Πίνακας 9.18 Συνδεδεμένα RoboCAPs του Προβλήματος 2.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

9.4.2.2 Λύση με χρήση του αλγορίθμου *Simulated Annealing*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 4976 ms.

Πίνακας 9.19 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.

BS	Location	RC Gateway
0	4	1

Πίνακας 9.20 Λεπτομέρειες χρηστών του Προβλήματος 2.

User	Location	Connected RC	Load
0	1	0	30.0
1	6	1	30.0

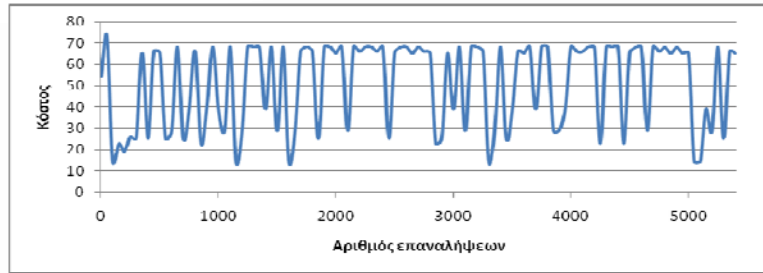
Πίνακας 9.21 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.

RC	Location	Capacity
0	3	300.0
1	4	300.0
2	5	300.0

Πίνακας 9.22 Συνδεδεμένα RoboCAPs του Προβλήματος 2.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

Παρατηρούμε πως και σε αυτήν την περίπτωση, όλα τα RoboCAPs είναι συνδεδεμένα μεταξύ τους.



Εικόνα 9.10 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 2.

9.4.2.3 Λύση με χρήση του αλγορίθμου *Simulated Annealing* με στοιχεία *ACO*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 5040 ms.

Πίνακας 9.23 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 2.

BS	Location	RC Gateway
0	4	1

Πίνακας 9.24 Λεπτομέρειες χρηστών του Προβλήματος 2.

User	Location	Connected RC	Load
0	1	2	30.0
1	6	1	30.0

Πίνακας 9.25 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 2.

RC	Location	Capacity
0	0	300.0
1	2	300.0
2	1	300.0

Πίνακας 9.26 Συνδεδεμένα RoboCAPs του Προβλήματος 2.

RC	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

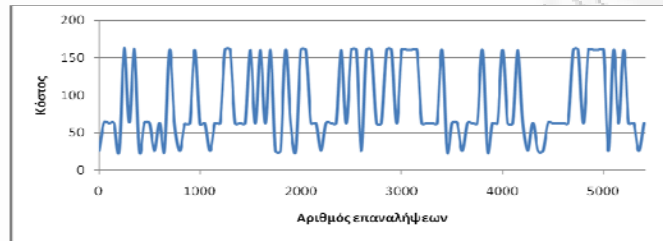
Όπως στις προηγούμενες περιπτώσεις, όλα τα RoboCAPs είναι συνδεδεμένα μεταξύ τους.

Πίνακας 9.27 Πίνακας φερομόνης του Προβλήματος 2.

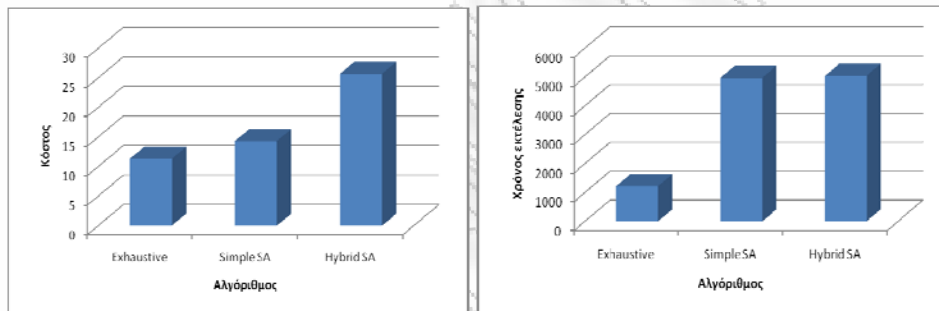
Nodes	0	1	2	3	4	5	6	7	8
0	0	64,24605	37,48181	0	0	0	0	0	0
1	64,24605	0	0	0	0	0	0	0	0
2	37,48181	0	0	37,48181	37,48181	0	0	0	0
3	0	0	37,48181	0	0	0	0	0	0
4	0	0	37,48181	0	0	37,51458	37,48181	37,48932	0
5	0	0	0	0	37,51458	0	0	0	0
6	0	0	0	0	37,48181	0	0	0	0
7	0	0	0	0	37,48932	0	0	0	37,48181
8	0	0	0	0	0	0	0	37,48181	0

Πίνακας 9.28 Πίνακας πιθανοτήτων του Προβλήματος 2.

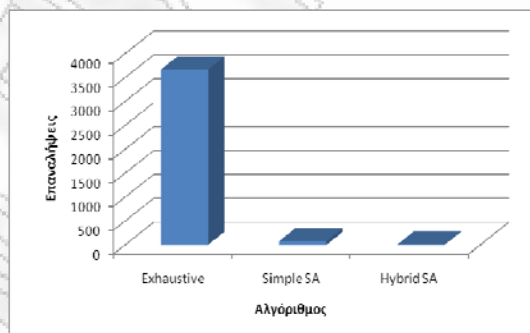
Nodes	0	1	2	3	4	5	6	7	8
0	0	0,631548	0,368452	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
2	0,333333	0	0	0,333333	0,333333	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0,249933	0	0	0,250151	0,249933	0,249983	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0,50005	0	0	0	0,49995
8	0	0	0	0	0	0	0	1	0



Εικόνα 9.11 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 2.



Εικόνα 9.12 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.



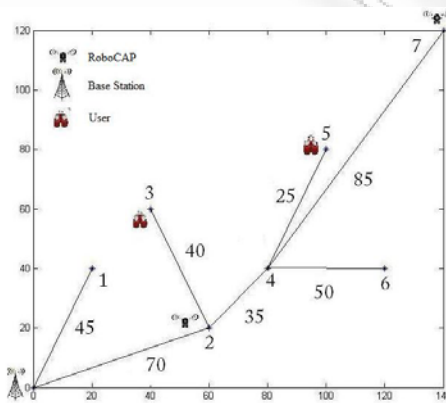
Εικόνα 9.13 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.

Από τα παραδείγματα παρατηρούμε πως όπως και στο προηγούμενο παράδειγμα, έτσι και τώρα ο απλός αλγόριθμος SA υπολογίζει ένα βέλτιστο κόστος που βρίσκεται πιο κοντά στην θεωρητική λύση που υπολογίζεται μέσω της εξαντλητικής αναζήτησης σε σχέση με τον υβριδικό αλγόριθμο SA-ACO που μας επιστρέφει μεγαλύτερη τιμή (Εικόνα 9.12(α)). Ο χρόνος εκτέλεσης (Εικόνα 9.12(β)) και των δύο αλγορίθμων είναι σχεδόν ίδιος όπως παρατηρείται. Όμως ο υβριδικός

αλγόριθμος και σε αυτή την περίπτωση χρειάζεται μικρότερο αριθμό επαναλήψεων (Εικόνα 9.13) για να επιτύχει το βέλτιστο κόστος του. Τέλος και σε αυτό το παράδειγμα, λόγω του μικρού μεγέθους του προβλήματος, η εξαντλητική αναζήτηση εκτελείται πολύ πιο γρήγορα, όμως χρειάζεται πολύ μεγάλο αριθμό επαναλήψεων μέχρι να υπολογίσει το πραγματικό βέλτιστο κόστος.

9.4.3 Παράδειγμα 3

Δίνεται ο γράφος όπως φαίνεται στην Εικόνα 9.14. Θεωρούμε ένα Σταθμό Βάσης στον κόμβο 0, δύο RoboCAPs στους κόμβους 2 και 7 με χωρητικότητα 300 Erlangs και εμβέλεια 200m και δύο χρήστες στους κόμβους 3 και 5 με φορτίο 80 Erlangs. Τα RoboCAPs επικοινωνούν με τους χρήστες καθώς και με άλλα RoboCAPs μέσω του πρωτοκόλλου 802.11n [59]. Αναζητούνται εκείνες οι θέσεις στον γράφο για τα RoboCAPs οι οποίες επιτυγχάνουν βέλτιστο κόστος.



Εικόνα 9.14 Γράφος του Προβλήματος 3.

9.4.3.1 Λύση μέσω εξαντλητικής αναζήτησης

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες. Ο χρόνος εκτέλεσης ήταν 192 ms.

Πίνακας 9.29 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.

BS	Location	RC Gateway
0	0	0

Πίνακας 9.30 Λεπτομέρειες χρηστών του Προβλήματος 3.

User	Location	Connected RC	Load
0	3	1	80.0
1	5	0	80.0

Πίνακας 9.31 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.

RC	Location	Capacity
0	1	300.0
1	3	300.0

Πίνακας 9.32 Συνδεδεμένα RoboCAPs του Προβλήματος 3.

RC	0	1
0	0	1
1	1	0

9.4.3.2 Λύση με χρήση του αλγορίθμου *Simulated Annealing*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 4822 ms.

Πίνακας 9.33 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.

BS	Location	RC Gateway
0	0	0

Πίνακας 9.34 Λεπτομέρειες χρηστών του Προβλήματος 3..

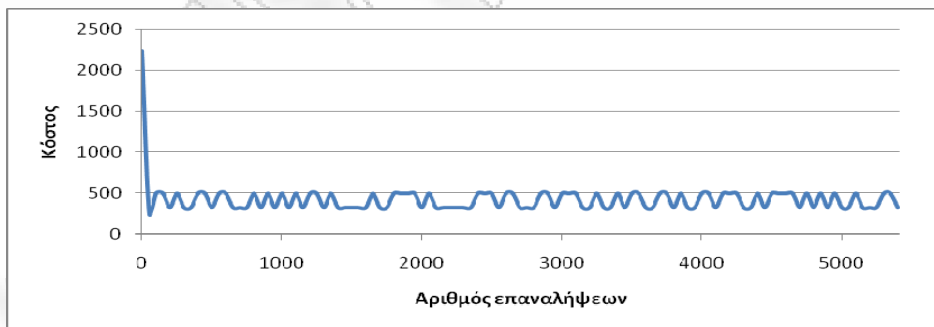
User	Location	Connected RC	Load
0	3	0	80.0
1	5	1	80.0

Πίνακας 9.35 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.

RC	Location	Capacity
0	1	300.0
1	4	300.0

Πίνακας 9.36 Συνδεδεμένα RoboCAPs του Προβλήματος 3.

RC	0	1
0	0	1
1	1	0



Εικόνα 9.15 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 3.

9.4.3.3 Λύση με χρήση του αλγορίθμου *Simulated Annealing* με στοιχεία *ACO*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και γραφήματα. Ο χρόνος εκτέλεσης ήταν 4744 ms.

Πίνακας 9.37 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 3.

BS	Location	RC Gateway
0	0	0

Πίνακας 9.38 Λεπτομέρειες χρηστών του Προβλήματος 3.

User	Location	Connected RC	Load
0	3	0	80.0
1	5	0	80.0

Πίνακας 9.39 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 3.

RC	Location	Capacity
0	0	300.0
1	0	300.0

Πίνακας 9.40 Συνδεδεμένα RoboCAPs του Προβλήματος 3.

RC	0	1
0	0	1
1	1	0

Πίνακας 9.41 Πίνακας φερομόνης του Προβλήματος 3.

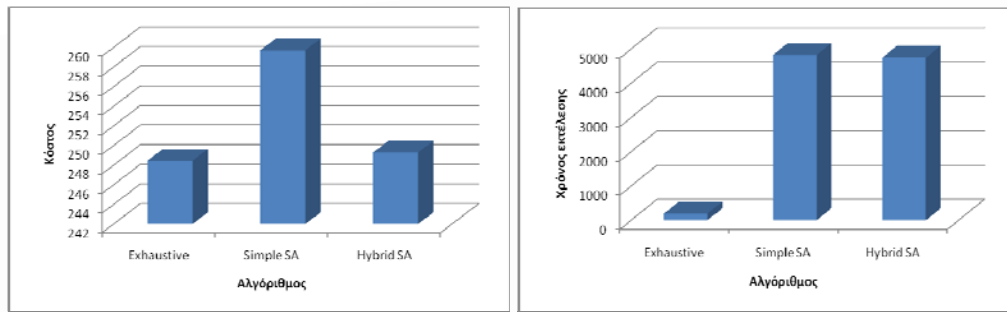
Nodes	0	1	2	3	4	5	6	7
0	0	16,0278772	10,066554	0	0	0	0	0
1	16,02788	0	0	0	0	0	0	0
2	10,06655	0	0	10,0654	10,06558	0	0	0
3	0	0	10,065396	0	0	0	0	0
4	0	0	10,06558	0	0	10,06467	10,065	10,06495
5	0	0	0	0	10,06467	0	0	0
6	0	0	0	0	10,065	0	0	0
7	0	0	0	0	10,06495	0	0	0

Πίνακας 9.42 Πίνακας πιθανοτήτων του Προβλήματος 3.

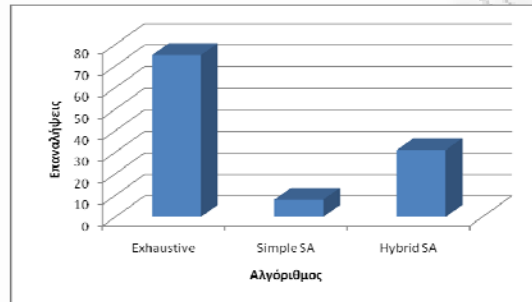
Nodes	0	1	2	3	4	5	6	7
0	0	0,614225957	0,385774	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	0,333357	0	0	0,333319	0,333325	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0,2500131	0	0	0,249991	0,249999	0,249998
5	0	0	0	0	1	0	0	0
6	0	0	0	0	1	0	0	0
7	0	0	0	0	1	0	0	0



Εικόνα 9.16 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 3.



Εικόνα 9.17 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.

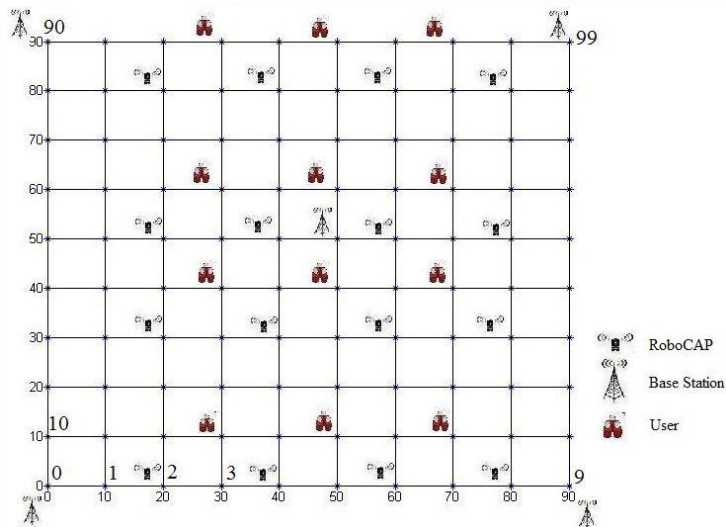


Εικόνα 9.18 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.

Παρατηρούμε πως σε αυτό το παράδειγμα, το κόστος που υπολογίζει ο υβριδικός αλγόριθμος συναρτήσεως του αριθμού των επαναλήψεων παρουσιάζει πολύ μικρές διακυμάνσεις (Εικόνα 9.16). Αυτό συμβαίνει διότι τα RoboCAPs ακολουθούν τα ίχνη της φερομόνης που βρίσκεται στις πλευρές του γράφου και οδηγούν σε λύσεις με μικρό κόστος. Επίσης το βέλτιστο κόστος (Εικόνα 9.17(α)) που υπολογίζει ο υβριδικός αλγόριθμος είναι πολύ μικρότερο σε σχέση με αυτό του απλού αλγορίθμου SA, και μάλιστα προσεγγίζει την θεωρητική λύση της εξαντλητικής αναζήτησης. Ακόμα, ο χρόνος εκτέλεσης (Εικόνα 9.17(β)) του υβριδικού αλγορίθμου είναι λίγο μικρότερος και σχεδόν ίσος με αυτόν του απλού SA. Όμως ο απλός SA χρειάστηκε λιγότερο αριθμό επαναλήψεων (Εικόνα 9.18) για να υπολογίσει το βέλτιστο κόστος του. Τέλος για ακόμα μια φορά, λόγω του μικρού μεγέθους του προβλήματος, η εξαντλητική αναζήτηση είναι η γρηγορότερη σε χρόνο εκτέλεσης μέθοδος, αλλά χρειάζεται μεγάλο αριθμό επαναλήψεων μέχρι να υπολογίσει τη θεωρητική λύση.

9.4.4 Παράδειγμα 4

Δίνεται ο γράφος όπως φαίνεται στην Εικόνα 9.19. Όλες οι πλευρές του γράφου έχουν κόστος μετακίνησης 10. Το πρόβλημα διαθέτει 5 Σταθμούς Βάσης, 16 RoboCAPs και 12 χρήστες στις θέσεις που φαίνονται στην Εικόνα. Η επικοινωνία μεταξύ των RoboCAPs και των χρηστών καθώς και των RoboCAPs μεταξύ τους γίνεται με το πρωτόκολλο 802.11n [59]. Αναζητείται η θέση των RoboCAPs που προσφέρει το βέλτιστο κόστος.



Εικόνα 9.19 Ο γράφος για το Πρόβλημα 4.

9.4.4.1 Λύση μέσω του αλγορίθμου *Simulated Annealing*

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και εικόνες. Ο χρόνος εκτέλεσης ήταν 13260 ms.

Πίνακας 9.43 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 4.

BS	Location	RC Gateway
0	0	0
1	9	3
2	55	10
3	90	12
4	99	15

Πίνακας 9.44 Λεπτομέρειες χρηστών του Προβλήματος 4.

User	Location	Connected RC	Load	User	Location	Connected RC	Load
0	13	0	10.0	6	63	12	10.0
1	15	1	10.0	7	65	13	10.0
2	17	3	10.0	8	67	7	10.0
3	43	12	10.0	9	93	12	10.0
4	45	15	10.0	10	95	14	10.0
5	47	7	10.0	11	97	15	10.0

Πίνακας 9.45 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 4.

RC	Location	Capacity	RC	Location	Capacity
0	45	500.0	8	55	500.0
1	35	500.0	9	55	500.0
2	54	500.0	10	54	500.0
3	46	500.0	11	44	500.0
4	34	500.0	12	44	500.0
5	45	500.0	13	55	500.0
6	44	500.0	14	54	500.0
7	45	500	15	45	500.0



Εικόνα 9.20 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 4.

9.4.4.2 Λύση μέσω του αλγορίθμου *Simulated Annealing* με στοιχεία ACO

Τα αποτελέσματα της προσομοίωσης παρατίθενται στους παρακάτω πίνακες και εικόνες. Ο χρόνος εκτέλεσης ήταν 105987 ms.

Πίνακας 9.46 Λεπτομέρειες των Σταθμών Βάσης του Προβλήματος 4.

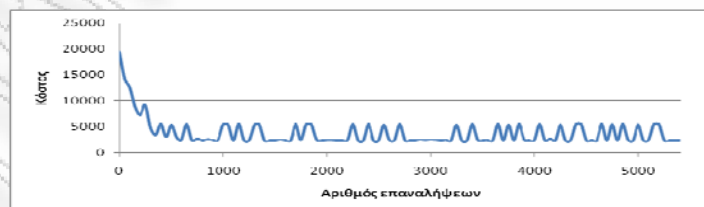
BS	Location	RC Gateway
0	0	0
1	9	3
2	55	10
3	90	12
4	99	15

Πίνακας 9.47 Λεπτομέρειες χρηστών του Προβλήματος 4.

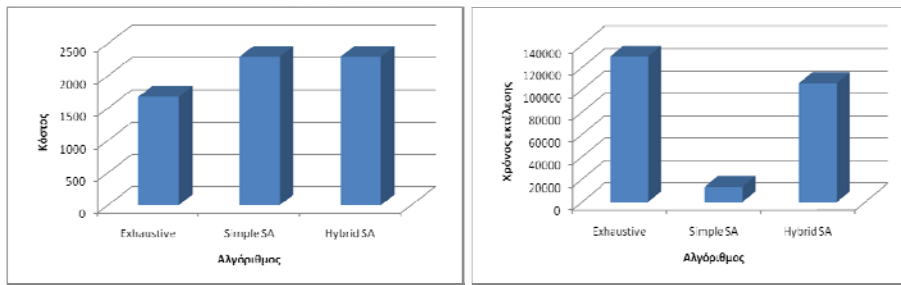
User	Location	Connected RC	Load	User	Location	Connected RC	Load
0	13	0	10.0	6	63	12	10.0
1	15	15	10.0	7	65	15	10.0
2	17	12	10.0	8	67	12	10.0
3	43	8	10.0	9	93	14	10.0
4	45	15	10.0	10	95	14	10.0
5	47	11	10.0	11	97	15	10.0

Πίνακας 9.48 Λεπτομέρειες για τα RoboCAPs του Προβλήματος 4.

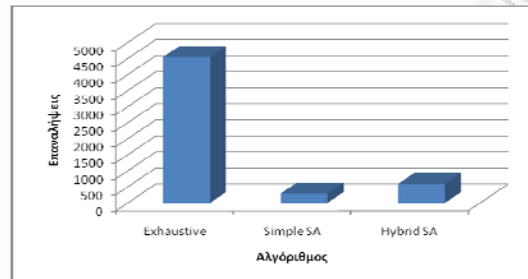
RC	Location	Capacity	RC	Location	Capacity
0	6	500.0	8	26	500.0
1	7	500.0	9	17	500.0
2	6	500.0	10	16	500.0
3	7	500.0	11	27	500.0
4	16	500.0	12	17	500.0
5	16	500.0	13	16	500.0
6	17	500.0	14	26	500.0
7	17	500.0	15	15	500.0



Εικόνα 9.21 Διάγραμμα Κόστους – Αριθμού Επαναλήψεων του Προβλήματος 4.



Εικόνα 9.22 (α) Βέλτιστο κόστος ανά αλγόριθμο (β) Χρόνος εκτέλεσης ανά αλγόριθμο.



Εικόνα 9.23 Ο αριθμός των επαναλήψεων που χρειάστηκε κάθε αλγόριθμος μέχρι την εύρεση του βέλτιστου κόστους.

Το συγκεκριμένο πρόβλημα αποτελεί ένα παράδειγμα μεγάλου μεγέθους. Οι δύο αλγόριθμοι υπολόγισαν σχεδόν το ίδιο βέλτιστο κόστος (Εικόνα 9.22(α)), με αυτό του υβριδικού αλγορίθμου SA-ACO να είναι ελαφρώς μικρότερο. Σε σχέση με τη θεωρητική λύση της εξαντλητικής αναζήτησης, το κόστος που υπολόγισαν οι δύο αλγόριθμοι είναι κατά 26,96% μεγαλύτερο. Όσον αφορά τον χρόνο εκτέλεσης (Εικόνα 9.22 (β)), ο απλός SA αποδείχθηκε ταχύτερος, αφού είναι ταχύτερος από την εξαντλητική αναζήτηση κατά 879,65% και κατά 699,3% από τον υβριδικό SA-ACO. Επίσης, ο απλός SA χρειάστηκε το μικρότερο αριθμό επαναλήψεων μέχρι να υπολογίσει το βέλτιστο κόστος του (Εικόνα 9.23). Σε σχέση με τον υβριδικό SA-ACO χρειάστηκε λιγότερες επαναλήψεις κατά 104,51%. Τέλος, παρατηρούμε πως σε αυτό το παράδειγμα που το μέγεθος του προβλήματος είναι μεγάλο, η εξαντλητική αναζήτηση χρειάζεται πολύ χρόνο για να ολοκληρωθεί με συνέπεια να είναι προτιμητέα μια μεταεωριστική μέθοδος.

9.5 Συμπεράσματα - Παρατηρήσεις

Σε αυτή την εργασία μελετήθηκε το πρόβλημα των Ρομποτικών, Γνωσιακών Σημείων Πρόσβασης (Robotic, Cognitive Access Points (RoboCAPs)). Τα RoboCAPs μπορούν να μετακινηθούν αυτόματα και αυτόνομα και να δημιουργήσουν ένα ράδιο-δίκτυο κάτω από δύσκολες συνθήκες ώστε να εξυπηρετήσουν τους χρήστες. Έμφαση δόθηκε στη βελτιστοποίηση της κατανομής των θέσεων τους σε ένα γράφο. Για τη μελέτη του προβλήματος, δημιουργήθηκε ένας υβριδικός αλγόριθμος που συνδυάζει τα καλύτερα χαρακτηριστικά δύο, γνωστών στον κόσμο των τηλεπικοινωνιών, αλγορίθμων, του Simulated Annealing και του Ant Colony

Optimization. Για να γίνουν μετρήσεις της συμπεριφοράς του αλγορίθμου, έγινε χρήση τεσσάρων σεναρίων κατά τη διάρκεια των οποίων μετρήθηκε τόσο το κόστος μιας αντικειμενικής συνάρτησης όσο και ο χρόνος εκτέλεσης και συγκρίθηκαν τόσο με τα αποτελέσματα του απλού Simulated Annealing, όσο και με αυτά της εξαντλητικής αναζήτησης.

Καθώς σε αυτή την εργασία έγινε η πρώτη υλοποίηση αυτού του υβριδικού αλγορίθμου, ο αλγόριθμος μπορεί να βελτιστοποιηθεί περαιτέρω. Στις περισσότερες περιπτώσεις παρατηρήθηκε πως ο υβριδικός αλγόριθμος χρειάστηκε τον μικρότερο αριθμό επαναλήψεων για να φτάσει στον υπολογισμό του βέλτιστου κόστους. Επομένως θέτοντας κάποιους επιπλέον περιορισμούς (π.χ. μικρότερο ρυθμό ψύξης (cooling rate)) θα μπορούσε να βελτιωθεί ο χρόνος εκτέλεσης. Ακόμα, μπορεί να χρησιμοποιηθεί κάποια καλύτερη εξίσωση για την τοποθέτηση φερομόνης στις πλευρές του γράφου, όπως και για την ανανέωση των τιμών της μετά το πέρας κάθε επανάληψης του αλγορίθμου, όπως συμβαίνει με τις διάφορες παραλλαγές των ACO αλγορίθμων.

Τέλος, όσον αφορά τα RoboCAPs, για να έχουμε μια εικόνα πιο κοντά στην πραγματικότητα, θα μπορούσαν να επιλέγουν δυναμικά ποιο πρωτόκολλο επικοινωνίας θα χρησιμοποιήσουν (π.χ. Wi-Fi, UMTS, ή ακόμα και δορυφορική επικοινωνία) ώστε να επιτύχουν ακόμα καλύτερο QoS. Επίσης, θα μπορούσαν να χρησιμοποιούν διαφορετικές συχνότητες ώστε να ελαττώνονται οι παρεμβολές λόγω χρήσης ίδιας συχνότητας.

10 Παράρτημα

Σε αυτό το τμήμα παρατίθεται ο πηγαίος κώδικας του απλού αλγορίθμου Simulated Annealing καθώς και του υβριδικού Simulated Annealing – Ant Colony Optimization.

10.1 Κλάση InitProblem

```
import java.io.*;

import com.rits.cloning.Cloner;

public class InitProblem implements Cloneable {

    int[][] RCConnect, prevRCConnect;

    RC[] Robo;

    BS[] Station;

    Users[] User;

    InitProblem(RC[] robo, BS[] station, Users[] user) {

        Cloner cloner = new Cloner();

        Robo = cloner.deepClone(robo);

        Station = cloner.deepClone(station);

        User = cloner.deepClone(user);

        RCConnect = new int[Lib.n_RC][Lib.n_RC];

        setConnect();
    }
}
```

```

}

double comCost(int n1, int n2) {    //calculates Tc using Free-Space Loss
    double dist = Lib.nodes[n1].distance(Lib.nodes[n2]);

    //double cost =
    20.0*Math.log10(dist)+20.0*Math.log10(Lib.Frequency)-147.56;

    double cost = Math.pow((4.0*Math.PI*dist*Lib.Frequency)/(3.0E8),2);
    return cost;
}

double euclCost(int n1, int n2) {
    return Lib.nodes[n1].distance(Lib.nodes[n2]);
}

double movCost(int n1, int n2) {
    return Lib.distance[n1][n2];
}

int[][] setRCConnect() {
    int[][] RCConnect = new int[Lib.n_RC][Lib.n_RC];

    for (int ri = 0; ri<Lib.n_RC; ri++) {
        for (int rj=0; rj<Lib.n_RC; rj++) {
            if (ri == rj) RCConnect[ri][rj] = 0;
            else {

```

```

        if
(Lib.nodes[Robo[ri].Location].distance(Lib.nodes[Robo[rj].Location]) <=
Robo[ri].Radius) {

                RCConnect[ri][rj] = 1;

                //System.out.println("RoboCAP " + ri + "
is now connected with RoboCAP " + rj);

                }

                else RCConnect[ri][rj] = 0;

        }

    }

    return RCConnect;

}

/*

boolean testRCLoc(int[] testedArray, int curRobo) {

    boolean flag = true;

    for (int r=0; r<testedArray.length; r++) {

        if ((r != curRobo) && (testedArray[r] == testedArray[curRobo]))

            flag = false;

    }

    return flag;

}

*/

```

```

void setUsersRCConnect() {

    double[][] curCost = new double[Lib.n_Users][Lib.n_RC];

    for (int u=0; u<Lib.n_Users; u++) {
        for (int r=0; r<Lib.n_RC; r++) {
            curCost[u][r] = comCost(User[u].Location,
Robo[r].Location);
        }
    }

    for (int u=0; u<Lib.n_Users; u++) {
        for (int r=0; r<Lib.n_RC; r++) {
            if ((curCost[u][r] <= User[u].bestCost)) {
                if ((User[u].bestRobo != r) && (Robo[r].Capacity
>= Robo[r].Load + User[u].Load)) {
                    User[u].bestCost = curCost[u][r];
                    if (User[u].prevBestRobo >= 0) {
                        Robo[User[u].prevBestRobo].Load -= User[u].Load;

                            for (int rc=0; rc<Lib.n_RC; rc++) {
                                if ((rc !=
User[u].prevBestRobo) && (RCConnect[rc][User[u].prevBestRobo] == 1))
                                    Robo[rc].Load -=
User[u].Load;
                            }
                }
            }
        }
    }
}

```



```
    }  
  
    User[u].prevBestRobo =  
  
    User[u].bestRobo = r;  
    Robo[r].Load += User[u].Load;  
    for (int rc=0; rc<Lib.n_RC; rc++) {  
        if ((r != rc) && (RCConnect[rc][r]  
== 1) && (Robo[rc].Capacity >= Robo[rc].Load + User[u].Load))  
            Robo[rc].Load +=  
User[u].Load;  
    }  
}  
  
else if ((User[u].bestRobo == r) &&  
(Robo[r].Capacity >= Robo[r].Load + User[u].Load)) {  
    User[u].bestCost = curCost[u][r];  
    User[u].prevBestRobo =  
User[u].bestRobo;  
    User[u].bestRobo = r;  
}  
}  
}  
}  
}
```

```
public void printArray(int[][] Array, String filename) {  
    try {
```

```
        BufferedWriter bw = new BufferedWriter(new  
        FileWriter(filename));
```

```
        for (int i=0; i<Array.length; i++) {  
            for (int j=0; j<Array[i].length; j++) {  
                bw.write(Array[i][j]+"\\t");  
            }  
            bw.write("\\r\\n");  
        }  
        bw.close();
```

```
    } catch (IOException e) { e.printStackTrace(); }
```

```
}
```

```
double UserCost() {
```

```
    double userCost = 0.0;  
    for (int u=0; u<Lib.n_Users; u++) {  
        userCost += User[u].bestCost;  
    }  
    return userCost/1E5;
```

```
}
```

```
double RCCost() {
```

```
    double RCCost = 0.0;  
    for (int ri=0; ri<Lib.n_RC; ri++) {  
        for (int rj=0; rj<Lib.n_RC; rj++) {  
            if (RCConnect[ri][rj] == 1) {
```

```

        if (Robo[ri].Location != Robo[rj].Location)
RCCost += comCost(Robo[ri].Location, Robo[rj].Location);

        else RCCost +=
1000*comCost(Robo[ri].Location, Robo[rj].Location);

    }

}

return RCCost/1E5;

}

```

```

void setBSRCCConnect() {

    double[][] curCost = new double[Lib.n_BS][Lib.n_RC];

    for (int b=0; b<Lib.n_BS; b++) {
        for (int r=0; r<Lib.n_RC; r++) {
            curCost[b][r] = comCost(Station[b].Location,
Robo[r].Location);
        }
    }

    for (int b=0; b<Lib.n_BS; b++) {
        for (int r=0; r<Lib.n_RC; r++) {
            if (curCost[b][r] < Station[b].bestCost) {
                Station[b].bestCost = curCost[b][r];
                Station[b].bestRobo = r;
            }
        }
    }
}

```

```

    }
}

double BSCost() {
    double bsCost = 0.0;
    for (int b=0; b<Lib.n_BS; b++) {
        bsCost += Station[b].bestCost;
    }
    return bsCost/1E4;
}

```

```

double movRCCost() {
    double totalMovCost = 0.0;
    for (int r=0; r<Lib.n_RC; r++) {
        totalMovCost += Robo[r].mCost;
    }
    return totalMovCost;
}

```

```

void setConnect() {
    setBSRCCConnect();
    setUsersRCCConnect();
    RCCConnect = setRCCConnect();
}

```

```

    }

    double Cost() {
        double ccost = 0.25*UserCost() + 0.25*RCCost() + 0.25*BSCost() +
0.25*movRCCost();
        //System.out.println("Current cost: " + ccost);
        return ccost;
    }

    void printRCLocs() {
        for (int r=0; r<Lib.n_RC; r++) {
            System.out.println("RC " + r + " is at node " +
Robo[r].Location);
        }
    }

    public Object clone() {
        try {
            return super.clone();
        }
        catch (CloneNotSupportedException e) {
            System.err.println("Cloning not supported");
            return null;
        }
    }
}

```

```
}
```

10.2 Πηγαίος κώδικας απλού αλγορίθμου SA

```
import java.io.*;

import java.util.ArrayList;

import java.util.Random;

import java.text.*;

import com.rits.cloning.*;

public class simpleAnnealing {

    InitProblem ip2, ip_best;

    ArrayList<InitProblem> solutions;

    InitProblem simple_Annealing(InitProblem ip_IS) {

        InitProblem ip_CBS, ip_NS;

        Random randomGenerator = new Random();

        double CT=100.0, T=100.0, Dcost=Double.MAX_VALUE, d=0.0;

        double c_factor=0.96;

        int iteration=3, max_it=100000;

        int counter=0;

        solutions = new ArrayList<InitProblem>();

        Cloner cloner = new Cloner();
```

```

ip_CBS = cloner.deepClone(ip_IS);

ip_best = cloner.deepClone(ip_IS);

NumberFormat df = new DecimalFormat("#.#####");

double bestCost = Double.MAX_VALUE;

try {
    BufferedWriter bw2 = new BufferedWriter(new
FileWriter("NScost_simple.txt"));
    bw2.write(df.format(ip_CBS.Cost()) + "\r\n");
    for (CT=T; CT>1E-30; CT*=c_factor) {
        for (int it=0; it<iteration; it++) {
            counter++;

ip_NS=cloner.deepClone(createNeighb(ip_CBS));
            ip_NS.setConnect();
            if (counter%50 == 0)
                bw2.write(df.format(ip_NS.Cost()) +
"\r\n");
            solutions.add(cloner.deepClone(ip_NS));

            Dcost=ip_NS.Cost()-ip_CBS.Cost();
            if (Dcost<0) {
                ip_CBS=cloner.deepClone(ip_NS);
                ip_CBS.setConnect();

```

```

        //ip_CBS.printRCLocs();
    }
    else {
        d=randomGenerator.nextDouble();
        if ((Math.exp(-Dcost/CT))>d) {
            ip_CBS=cloner.deepClone(ip_NS);
            ip_CBS.setConnect();
            //ip_CBS.printRCLocs();
        }
    }
}
}
}
if (counter > max_it) {
    System.out.println("Reached " + max_it + "
iterations.");
    break;
}
}
}
bw2.close();
} catch (IOException e) { e.printStackTrace(); }
int pointer = 0;
for (int s=0; s<solutions.size(); s++) {
    solutions.get(s).setConnect();
    if (solutions.get(s).Cost() < bestCost) {

```



```

        bestCost = solutions.get(s).Cost();
        pointer = s;
        ip_best = cloner.deepClone(solutions.get(s));
    }
}

System.out.println("End of Simulated Annealing");
System.out.println("Temperature: " + CT);
System.out.println("Iterations: " + counter);
System.out.println("Best cost of Simulated Annealing:" +
ip_best.Cost());
System.out.println("Best solution found at place: " + pointer + "/" +
counter);

//Printing statistics
ip_best.printArray(ip_best.RCConnect, "RCConnect_simple.txt");
try {
    BufferedWriter bws = new BufferedWriter(new
FileWriter("statistics_simple.txt"));
    bws.write("BS" + "\t" + "Location" + "\t" + "RC Gateway"
+"\r\n");
    for (int b=0; b<Lib.n_BS; b++) {
        bws.write(b + "\t" + ip_best.Station[b].Location + "\t"
+ ip_best.Station[b].bestRobo + "\r\n");
    }
    bws.write("User" + "\t" + "Location" + "\t" + "Connected RC" +
"\t" + "Load" + "\r\n");
}

```

```

        for (int u=0; u<Lib.n_Users; u++) {
            bws.write(u + "\t" + ip_best.User[u].Location + "\t" +
ip_best.User[u].bestRobo + "\t" + ip_best.User[u].Load + "\r\n");
        }
        bws.write("RC" + "\t" + "Location" + "\t" + "Load" + "\t" +
"Capacity" + "\r\n");
        for (int r=0; r<Lib.n_RC; r++) {
            bws.write(r + "\t" + ip_best.Robo[r].Location + "\t" +
ip_best.Robo[r].Load + "\t" + ip_best.Robo[r].Capacity + "\r\n");
        }
        bws.close();
    }
    catch (IOException e) { e.printStackTrace(); }

    return cloner.deepClone(ip_CBS);
}

```

```

InitProblem createNeighb(InitProblem ip1) {
    Cloner cloner = new Cloner();
    ip2 = cloner.deepClone(ip1);
    Random generator = new Random();
    int randRC;
    int prevLoc, newLoc, randLoc;
    randRC = generator.nextInt(Lib.n_RC);
    prevLoc = ip2.Robo[randRC].Location;
}

```

```

        randLoc = generator.nextInt(Lib.adjList[prevLoc].size());
        newLoc = Lib.adjList[prevLoc].get(randLoc);
        ip2.Robo[randRC].mCost = Lib.bestDist[prevLoc][newLoc];
        ip2.Robo[randRC].Location = newLoc;
        //System.out.println("RoboCAP " + randRC + " is moving to Node " +
newLoc);

        ip2.setConnect();

        return cloner.deepClone(ip2);

    }

```

boolean isOccupied(int node, int curRobo) { //checks if there is another RC at the node

```

        boolean flag = false;
        for (int r=0; r<Lib.n_RC; r++) {
            if (r == curRobo) continue;
            else {
                if (ip2.Robo[r].Location == node)
                    flag = true;
            }
        }
        return flag;
    }

```

```

boolean checkHalt(int curRobo) {
    boolean flag = false;
    int curLoc = ip2.Robo[curRobo].Location;
    if (Lib.adjList[curLoc].size() == 1) {
        for (int r=0; r<Lib.n_RC; r++) {
            if (ip2.Robo[r].Location == Lib.adjList[curLoc].get(0))
                flag = true;
        }
    }
    return flag;
}
}

```

10.3 Πηγαίος κώδικας υβριδικού SA-ACO

```

import java.io.*;
import java.util.ArrayList;
import java.util.Random;
import java.text.*;
import com.rits.cloning.*;

public class SimAnnealing {

    int counter=0, pointer=0;

    InitProblem ip2, ip_best;

    NumberFormat df = new DecimalFormat("#.#####");

```

```

ArrayList<InitProblem> solutions;

InitProblem Sim_Annealing(InitProblem ip_IS) {

    initPheromones(ip_IS.Cost());
    updateProbTable();
    InitProblem ip_CBS, ip_NS;
    Random randomGenerator = new Random();
    double CT=100.0, T=100.0, Dcost=Double.MAX_VALUE, d=0.0;
    double c_factor=0.96;
    ip_best = null;
    int iteration=3, max_it=100000;
    solutions = new ArrayList<InitProblem>();
    Cloner cloner = new Cloner();

    ip_CBS = cloner.deepClone(ip_IS);
    double bestCost = Double.MAX_VALUE;

    try {
        BufferedWriter bw2 = new BufferedWriter(new
FileWriter("NScost_aco.txt"));
        bw2.write(df.format(ip_CBS.Cost()) + "\r\n");
        for (CT=T; CT>1E-30; CT*=c_factor) {
            for (int it=0; it<iteration; it++) {
                counter++;
            }
        }
    }
}

```

```

ip_NS=cloner.deepClone(createPherProbNeighb(ip_CBS));

        ip_NS.setConnect();
        solutions.add(cloner.deepClone(ip_NS));
        if (counter%50 == 0)
            bw2.write(df.format(ip_NS.Cost()) +
"\r\n");

        Dcost=ip_NS.Cost()-ip_CBS.Cost();
        if (Dcost<0) {
            ip_CBS=cloner.deepClone(ip_NS);
            ip_CBS.setConnect();

            //ip_CBS.printRCLocs();
        }
        else {
            d=randomGenerator.nextDouble();
            if ((Math.exp(-Dcost/CT))>d) {
                ip_CBS=cloner.deepClone(ip_NS);

                ip_CBS.setConnect();
                //ip_CBS.printRCLocs();
            }
        }
    }
}

```

```

        if (counter > max_it) {
            System.out.println("Reached " + max_it + "
iterations.");
            break;
        }
    }
    bw2.close();
} catch (IOException e) { e.printStackTrace(); }

int pointer = 0;
for (int s=0; s<solutions.size(); s++) {
    solutions.get(s).setConnect();
    if (solutions.get(s).Cost() < bestCost) {
        bestCost = solutions.get(s).Cost();
        pointer = s;
        ip_best = cloner.deepClone(solutions.get(s));
    }
}

System.out.println("End of ACO Simulated Annealing");
System.out.println("Temperature: " + CT);
System.out.println("Iterations: " + counter);
System.out.println("Best cost of ACO Simulated Annealing:" +
ip_best.Cost());
System.out.println("Best solution found at place: " + pointer + "/" +
counter);

```

```

//Printing statistics

ip_best.printArray(ip_best.RCConnect, "RCConnect_aco.txt");

try {

    BufferedWriter bws = new BufferedWriter(new
FileWriter("statistics_aco.txt"));

    bws.write("BS" + "\t" + "Location" + "\t" + "RC Gateway"
+"\r\n");

    for (int b=0; b<Lib.n_BS; b++) {

        bws.write(b + "\t" + ip_best.Station[b].Location + "\t"
+ ip_best.Station[b].bestRobo + "\r\n");

    }

    bws.write("User" + "\t" + "Location" + "\t" + "Connected RC" +
"\t" + "Load" + "\r\n");

    for (int u=0; u<Lib.n_Users; u++) {

        bws.write(u + "\t" + ip_best.User[u].Location + "\t" +
ip_best.User[u].bestRobo + "\t" + ip_best.User[u].Load + "\r\n");

    }

    bws.write("RC" + "\t" + "Location" + "\t" + "Load" + "\t" +
"Capacity" + "\r\n");

    for (int r=0; r<Lib.n_RC; r++) {

        bws.write(r + "\t" + ip_best.Robo[r].Location + "\t" +
ip_best.Robo[r].Load + "\t" + ip_best.Robo[r].Capacity + "\r\n");

    }

    bws.close();

}

catch (IOException e) { e.printStackTrace(); }

```



```

printPherProb();

return cloner.deepClone(ip_best);

}

InitProblem createPherProbNeighb(InitProblem ip1) {
    Cloner cloner = new Cloner();
    double init_cost;
    ip2 = cloner.deepClone(ip1);
    Random generator = new Random();
    int randRC=0, newLoc=0, prevLoc=0;
    double rand;
    randRC = generator.nextInt(Lib.n_RC);
    prevLoc = ip2.Robo[randRC].Location;
    init_cost = ip2.Cost();
    if (Lib.adjList[prevLoc].size() > 1) {
        rand = generator.nextDouble();
        double[] probArray = new double[Lib.adjList[prevLoc].size()];
        for (int n=0; n<probArray.length; n++) {
            probArray[n] =
Lib.probTable[prevLoc][Lib.adjList[prevLoc].get(n)];
        }
        for (int n=1; n<probArray.length; n++)
            probArray[n] += probArray[n-1];
    }
}

```

```

        for (int n=0; n<probArray.length-1; n++) { //choose new
location
            if (n == 0) {
                if (rand < probArray[n]) newLoc =
Lib.adjList[prevLoc].get(n);
            }
            else if (n == probArray.length-2) {
                if (rand >= probArray[n]) newLoc =
Lib.adjList[prevLoc].get(n);
            }
            else {
                if ((rand >= probArray[n-1]) && (rand <
probArray[n])) newLoc = Lib.adjList[prevLoc].get(n);
            }
        }
    }
    else newLoc = Lib.adjList[prevLoc].get(0);
    ip2.Robo[randRC].mCost = Lib.bestDist[prevLoc][newLoc];
    ip2.Robo[randRC].Location = newLoc;
    //System.out.println("RC" + randRC + " moved to node " + newLoc);
    ip2.setConnect();
    Lib.pherTable[prevLoc][newLoc] += 2.0/(init_cost+ip2.Cost());
    Lib.pherTable[newLoc][prevLoc] += 2.0/(init_cost+ip2.Cost());
    for (int ni=0; ni<Lib.n_Nodes; ni++) { //adding pheromone to other
edges to avoid overload
        for (int nj=0; nj<Lib.n_Nodes; nj++) {
            if (((ni==prevLoc) && (nj==newLoc)) || ((ni==newLoc)
&& (nj==prevLoc))) continue;

```

```

else {
    if (isNeighbor(ni,nj))
        Lib.pherTable[ni][nj] +=
1.0/(init_cost+ip2.Cost());
    }
}
}
checkPherTable();
updateProbTable();

return cloner.deepClone(ip2);
}

void checkPherTable() {
    for (int ni=0; ni<Lib.n_Nodes; ni++) {
        for (int nj=0; nj<Lib.n_Nodes; nj++) {
            if (!isNeighbor(ni,nj)) Lib.pherTable[ni][nj] = 0.0;
        }
    }
}

boolean isNeighbor(int n1, int n2) { //checks if node n1 is a neighbor of node
n2
    boolean flag = false;
    for (int i=0; i<Lib.adjList[n1].size(); i++) {
        if (n2 == Lib.adjList[n1].get(i))

```

```

        flag = true;
    }
    return flag;
}

void initPheromones(double cost) {
    for (int ni=0; ni<Lib.n_Nodes; ni++) {
        for (int nj=0; nj<Lib.n_Nodes; nj++) {
            if (isNeighbor(ni,nj))
                Lib.pherTable[ni][nj] = 10.0/cost;
        }
    }
}

```

```

boolean checkHalt(int curRobo) {
    boolean flag = false;
    int counter =0;
    int curLoc = ip2.Robo[curRobo].Location;
    for (int n=0; n<Lib.adjList[curLoc].size(); n++) {
        for (int r=0; r<Lib.n_RC; r++) {
            if (ip2.Robo[r].Location == Lib.adjList[curLoc].get(n))
                ++counter;
        }
    }
    if (counter == Lib.adjList[curLoc].size()) flag = true;
}

```

```
    return flag;
}
```

boolean isOccupied(int node, int curRobo) { //checks if there is another RC at the node

```
    boolean flag = false;
    for (int r=0; r<Lib.n_RC; r++) {
        if (r == curRobo) continue;
        else {
            if (ip2.Robo[r].Location == node)
                flag = true;
        }
    }
    return flag;
}
```

```
boolean checkForOtherRC(int curRobo) {
    boolean flag = false;
    int curLoc = ip2.Robo[curRobo].Location;
    for (int r=0; r<Lib.n_RC; r++) {
        if (r == curRobo) continue;
        else {
            if (ip2.Robo[r].Location == curLoc)
                flag = true;
        }
    }
}
```

```

        return flag;
    }

    void updateProbTable() {
        double[] sumPher = new double[Lib.n_Nodes];
        for (int ni=0; ni<Lib.n_Nodes; ni++) {
            for (int nj=0; nj<Lib.n_Nodes; nj++) {
                sumPher[ni] += Lib.pherTable[ni][nj];
            }
        }
        for (int ni=0; ni<Lib.n_Nodes; ni++) {
            for (int nj=0; nj<Lib.n_Nodes; nj++) {
                Lib.probTable[ni][nj] =
Lib.pherTable[ni][nj]/sumPher[ni];
            }
        }
    }

    void printPherProb() {
        try {
            BufferedWriter bwp = new BufferedWriter(new
FileWriter("pheromones.txt"));

            BufferedWriter bwpr = new BufferedWriter(new
FileWriter("probabilities.txt"));

            for (int ni=0; ni<Lib.n_Nodes; ni++) {
                for (int nj=0; nj<Lib.n_Nodes; nj++) {

```

```
        bwp.write(df.format(Lib.pherTable[ni][nj]) +
"\t");
        bwpr.write(df.format(Lib.probTable[ni][nj]) +
"\t");
    }
    bwp.write("\r\n");
    bwpr.write("\r\n");
}
bwp.close();
bwpr.close();
}
catch (IOException e) { e.printStackTrace(); }
}
}
```

РАНЕЕЗНАМО ПЕРПАА

11 Αναφορές - Βιβλιογραφία

- [1] MAETERLINCK, M. THE LIFE OF THE WHITE ANT. LONDON: GEORGE ALLEN AND UNWIN, 1927.
- [2] HOLLDOBLER, B., AND E. O. WILSON. THE ANTS. CAMBRIDGE, MA: HARVARD UNIVERSITY PRESS, 1990.
- [3] HOLLDOBLER, B., AND E. O. WILSON. "THE MULTIPLE RECRUITMENT SYSTEMS OF THE AFRICAN WEAVER ANT OECOPHYLLA LONGINODA (LATREILLE)." BEHAV. ECOL. SOCIOBIOL. 3 (1978): 19-60.
- [4] BURTON, J. L., AND N. R. FRANKS. "THE FORAGING ECOLOGY OF THE ARMY ANT.
- [5] ECITON RAPAX: AN ERGONOMIC ENIGMA?" ECOL. ENTOMOL. 10 (1985): 131-141.
- [6] RETTENMEYER, C. W. "BEHAVIORAL STUDIES OF ARMY ANTS." UNIV. KANS. SCI. BULL. 44 (1963): 281-465.
- [7] SCHNEIRLA, T. C. "ARMY ANTS." IN ARMY ANTS: A STUDY IN SOCIAL ORGANIZATION, EDITED BY H. R. TOPOFF. SAN FRANCISCO, CA: W. H. FREEMAN, 1971.
- [8] JEANNE, R. L. "THE EVOLUTION OF THE ORGANIZATION OF WORK IN SOCIAL INSECTS." MONIT. ZOOL. HAL. 20 (1986): 119-133.
- [9] ROBINSON, G. E. "REGULATION OF VISION OF LABOR IN INSECT SOCIETIES." ANNU. REV. ENTOMOL. 37 (1992): 637-665.
- [10] HAKEN, H. SYNERGETICS. BERLIN: SPRINGER-VERLAG, 1983.
- [11] NICOLIS, G., AND I. PRIGOGINE. SELF-ORGANIZATION IN NON-EQUILIBRIUM SYSTEMS. NEW YORK, NY: WILEY & SONS, 1977.
- [12] DENEUBOURG, J.-L., S. GOSS, N. R. FRANKS, AND J. M. PASTEELS. "THE BLIND LEADING THE BLIND: MODELLING CHEMICALLY MEDIATED ARMY ANT RAID PATTERNS." J. INSECT BEHAV. 2 (1989): 719-725.
- [13] BENI, G. "THE CONCEPT OF CELLULAR ROBOTIC SYSTEM." 'IN PROCEEDINGS 1988 IEEE INT. SYMP. ON INTELLIGENT CONTROL, 57-62. LOS ALAMITOS, CA: IEEE COMPUTER SOCIETY PRESS, 1988.
- [14] BENI, G., AND J. WANG. "SWARM INTELLIGENCE." IN PROCEEDINGS SEVENTH ANNUAL MEETING OF THE ROBOTICS SOCIETY OF JAPAN, 425-428. TOKYO: RSJ PRESS, 1989.
- [15] BENI, G., AND J. WANG. "THEORETICAL PROBLEMS FOR THE REALIZATION OF DISTRIBUTED ROBOTIC SYSTEMS." IN PROCEEDINGS 1991 IEEE INTERNATIONAL CONFERENCE ON ROBOTIC AND AUTOMATION, 1914-1920. LOS ALAMITOS, CA: IEEE COMPUTER SOCIETY PRESS, 1991.
- [16] E. BONABEAU, M. DORIGO, G. THERAULAZ, "SWARM INTELLIGENCE: FROM NATURAL TO ARTIFICIAL SYSTEMS".
- [17] BUTRIMENKO, A. V. "ON THE SEARCH FOR OPTIMAL ROUTES IN CHANGING GRAPHS." IZV. AKAD. NAUK SSSR SER. TEKHN. KIBERN. 6 (1964).
- [18] TSETLIN, M. L. AUTOMATON THEORY AND MODELLING OF BIOLOGICAL SYSTEMS. NEW YORK, NY: ACADEMIC PRESS, 1973.
- [19] STEFANYUK, V. L. "ON MUTUAL ASSISTANCE IN THE COLLECTIVE OF RADIOSTATIONS." INFORMATION TRANSMISSION PROBLEMS 7 (1971): 103-107.

- [20] M. DORIGO, M. BIRATTARI, T. STUTZLE, "ANT COLONY OPTIMIZATION: ARTIFICIAL ANTS AS A COMPUTATIONAL INTELLIGENCE TECHNIQUE", IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, NOVEMBER 2006.
- [21] P.-P. GRASSE, LES INSECTES DANS LEUR UNIVERS, PARIS, FRANCE: ED. DU PALAIS DE LA D'ÉCOUVERTE, 1946.
- [22] P.-P. GRASSE, "LA RECONSTRUCTION DU NID ET LES COORDINATIONS INTERINDIVIDUELLES CHEZ BELLICOSITERMES NATALENSIS ET CUBITERMES SP. LA THÉORIE DE LA STIGMERGIE: ESSAI D'INTERPRETATION DU COMPORTEMENT DES TERMITES CONSTRUCTEURS," INSECTES SOCIAUX, VOL. 6, PP. 41–81, 1959.
- [23] J.-L. DENEUBOURG, S. ARON, S. GOSS, AND J.-M. PASTEELS, "THE SELF-ORGANIZING EXPLORATORY PATTERN OF THE ARGENTINE ANT," JOURNAL OF INSECT BEHAVIOR, VOL. 3, P. 159, 1990.
- [24] S. GOSS, S. ARON, J.-L. DENEUBOURG, AND J.-M. PASTEELS, "SELF-ORGANIZED SHORTCUTS IN THE ARGENTINE ANT," NATURWISSENSCHAFTEN, VOL. 76, PP. 579–581, 1989.
- [25] M. DORIGO, V. MANIEZZO, AND A. COLORNI, "ANT SYSTEM: OPTIMIZATION BY A COLONY OF COOPERATING AGENTS," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B, VOL. 26, NO. 1, PP. 29–41, 1996.
- [26] M. DORIGO AND G. DI CARO, "THE ANT COLONY OPTIMIZATION META-HEURISTIC," IN NEW IDEAS IN OPTIMIZATION, D. CORNE ET AL., EDs., MCGRAW HILL, LONDON, UK, PP. 11–32, 1999.
- [27] M. DORIGO, G. DI CARO, AND L.M. GAMBARDILLA, "ANT ALGORITHMS FOR DISCRETE OPTIMIZATION," ARTIFICIAL LIFE, VOL. 5, NO. 2, PP. 137–172, 1999.
- [28] V. CERNY, "A THERMODYNAMICAL APPROACH TO THE TRAVELING SALESMAN PROBLEM," JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS, VOL. 45, NO. 1, PP. 41–51, 1985.
- [29] S. KIRKPATRICK, C.D. GELATT JR., AND M.P. VECCHI, "OPTIMIZATION BY SIMULATED ANNEALING," SCIENCE, VOL. 220, PP. 671–680, 1983.
- [30] F. GLOVER, "TABU SEARCH—PART I," ORSA JOURNAL ON COMPUTING, VOL. 1, NO. 3, PP. 190–206, 1989.
- [31] F. GLOVER, "TABU SEARCH—PART II," ORSA JOURNAL ON COMPUTING, VOL. 2, NO. 1, PP. 4–32, 1990.
- [32] F. GLOVER AND M. LAGUNA, TABU SEARCH, KLUWER ACADEMIC PUBLISHERS, 1997.
- [33] H.R. LOURENÇO, O. MARTIN, AND T. STUTZLE, "ITERATED LOCAL SEARCH," IN HANDBOOK OF METAHEURISTICS, SER. INTERNATIONAL SERIES IN OPERATIONS RESEARCH & MANAGEMENT SCIENCE, F. GLOVER AND G. KOCHENBERGER, EDs., KLUWER ACADEMIC PUBLISHERS, VOL. 57, PP. 321–353, 2002.
- [34] L.J. FOGEL, A.J. OWENS, AND M.J. WALSH, ARTIFICIAL INTELLIGENCE THROUGH SIMULATED EVOLUTION, JOHN WILEY & SONS, 1966.
- [35] J. HOLLAND, ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS, ANN ARBOR: UNIVERSITY OF MICHIGAN PRESS, 1975.
- [36] I. RECHENBERG, EVOLUTIONSSTRATEGIE—OPTIMIERUNG TECHNISCHER SYSTEME NACH PRINZIPIEN DER BIOLOGISCHEN INFORMATION, FROMMAN VERLAG, FREIBURG, GERMANY, 1973.
- [37] H.-P. SCHWEFEL, NUMERICAL OPTIMIZATION OF COMPUTER MODELS. JOHN WILEY & SONS, 1981.

- [38] M. DORIGO AND T. STUTZLE, ANT COLONY OPTIMIZATION, MIT PRESS, CAMBRIDGE, MA, 2004.
- [39] KIRKPATRICK, S., GELATT, JR., C.D, AND VECCHI, M.P. (1983), "OPTIMIZATION BY SIMULATED ANNEALING", SCIENCE 220, 671-680.
- [40] N. METROPOLIS, A. ROSENBLUTH, M, ROSENBLUTH, A. TELLER, AND E. TELLER, J. CHEM. PHYS. 21:1087-1092 (1953).
- [41] OZAN, K. TONGUZ, GIANLUIGI FERRARI (MAY 2006). JOHN WILEY & SONS.. ED. AD HOC WIRELESS NETWORKS: A COMMUNICATION-THEORETIC PERSPECTIVE.
- [42] ROYER, E., TOH, C. (APRIL 1999). "A REVIEW OF CURRENT ROUTING PROTOCOLS FOR AD HOC MOBILE WIRELESS NETWORKS". IEEE PERSONAL COMMUNICATIONS 6 (2): 46-55.
- [43] C K TOH (JANUARY 2002). PRENTICE HALL PUBLISHERS. ED. AD HOC MOBILE WIRELESS NETWORKS: PROTOCOLS & SYSTEMS. ISBN 0130078174.
- [44] FP6/IST PROJECT E2R (END-TO-END RECONFIGURABILITY). <http://www.e2r2.motlabs.com>, 2007.
- [45] WIRELESS WORLD RESEARCH FORUM (WWRF)). [HTTP://WWW.WIRELESS-WORLDRESEARCH.ORG](http://www.wireless-worldresearch.org), 2007.
- [46] W. HASSELBRING AND R. REUSSNER. TOWARD TRUSTWORTHY SOFTWARE SYSTEMS. COMPUTER, 39(4):91-92, 2006.
- [47] P.DEMESTICHAS, G.DIMITRAKOPOULOS, K.TSAGKARIS, V.STAVROULAKI, A.KATIDIOTIS, "INTRODUCING COGNITIVE SYSTEMS TO THE B3G WIRELESS WORLD", COGNITIVE WIRELESS NETWORKS: CONCEPTS, METHODOLOGIES AND VISIONS INSPIRING THE AGE OF ENLIGHTENMENT OF WIRELESS COMMUNICATIONS, PP. 253-269, SPRINGER, DORDRECHT, THE NETHERLANDS, 2007.
- [48] P. DEMESTICHAS, D. BOSCOVIC, V. STAVROULAKI, A. LEE, AND J. STRASSNER. M@ANGEL: AUTONOMIC MANAGEMENT PLATFORM FOR SEAMLESS WIRELESS COGNITIVE CONNECTIVITY TO THE MOBILE INTERNET. IEEE COMMUN. MAG., 44:118-127, JUNE 2006.
- [49] R. THOMAS, L. DASILVA, AND A. MACKENZIE. COGNITIVE NETWORKS. IN 1ST IEEE SYMPOSIUM ON DYNAMIC SPECTRUM ACCESS NETWORKS 2005 (DYSPAN 2005), PAGES 352-360, BALTIMORE, USA, NOVEMBER 2005.
- [50] J. STRASSNER, EDITOR. POLICY-BASED NETWORK MANAGEMENT: SOLUTIONS FOR THE NEXT GENERATION. MORGAN KAUFMANN, AUGUST 2003. ISBN 1-5586-0859-1.
- [51] T. MITCHEL, EDITOR. MACHINE LEARNING. ISBN 0-0704-2807-7. MCGRAW-HILL SCIENCE/ENGINEERING/MATH, MARCH 1997.
- [52] R. E. NEAPOLITAN, EDITOR. LEARNING BAYESIAN NETWORKS. ISBN 0-1301-2534-2. PRENTICE HALL, APRIL 2003.
- [53] P. DEMESTICHAS, G. DIMITRAKOPOULOS, J. STRASSNER, AND D. BOURSE. INTRODUCING RECONFIGURABILITY AND COGNITIVE NETWORKS CONCEPTS IN THE WIRELESS WORLD. VEHICULAR TECHNOLOGY MAGAZINE, IEEE, 1(2):32-39, 2006.
- [54] R. MATHAR AND T. NIESSEN, "OPTIMUM POSITIONING OF BASE STATIONS FOR CELLULAR RADIO NETWORKS," WIRELESS NETWORKS, VOL. 6, NO. 6, PP. 421-428, 2000.
- [55] S. HURLEY, "PLANNING EFFECTIVE CELLULAR MOBILE RADIO NETWORKS," IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 51, NO. 2, PP. 243-253, 2002.
- [56] K. TUTSCHKU, "DEMAND-BASED RADIO NETWORK PLANNING OF CELLULAR MOBILE COMMUNICATION SYSETMS," PROC. 17TH ANNUAL INFOCOM, 1998.

- [57] "NETWORKED ROBOTICS: ISSUES, ARCHITECTURES AND APPLICATIONS," [HTTP://WWW.ROBOTICS.READING.AC.UK/IROS04/](http://www.robotics.reading.ac.uk/iros04/), 2004.
- [58] "IEEE 802.11G-2003: FURTHER HIGHER DATA RATE EXTENSION IN THE 2.4 GHz BAND", IEEE, OCTOBER 2003.
- [59] "IEEE 802.11N-2009—AMENDMENT 5: ENHANCEMENTS FOR HIGHER THROUGHPUT", IEEE-SA, OCTOBER 2009
- [60] MIGLAYOUT - THE JAVA LAYOUT MANAGER FOR SWING, SWT AND JAVAFX, AVAILABLE ONLINE AT [HTTP://WWW.MIGLAYOUT.COM](http://www.miglayout.com), DECEMBER 2009.
- [61] JAVA DEEP-CLONING LIBRARY, AVAILABLE ONLINE AT [HTTP://ROBUST-IT.CO.UK/CLONE/INDEX.PHP](http://robust-it.co.uk/clone/index.php), DECEMBER 2009

РАНЕЕЗНАМО ПЕРПАА

РАНЕЕЗНАМО ПЕРПАА