



Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών
Τηλεπικοινωνιών και Δικτύων
Πολυτεχνική Σχολή Πανεπιστημίου Θεσσαλίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ACO: Αλγόριθμος εμπνευσμένος από την φύση για ad-hoc δίκτυα και clouds

Σχεδιασμός και υλοποίηση συστήματος προσομοίωσης
βασισμένο στην οργάνωση και την δομή που αναπτύσσουν οι
αποικίες μυρμηγκιών

ΖΥΓΟΥΡΑ Μ. ΒΑΪΑ

Βόλος, Μάρτιος 2013



Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών
Τηλεπικοινωνιών και Δικτύων
Πολυτεχνική Σχολή Πανεπιστημίου Θεσσαλίας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ACO: Αλγόριθμος εμπνευσμένος από την φύση για ad-hoc δίκτυα και clouds

Σχεδιασμός και υλοποίηση συστήματος προσομοίωσης
βασισμένο στην οργάνωση και την δομή που αναπτύσσουν οι
αποικίες μυρμηγκιών

ΖΥΓΟΥΡΑ Μ. ΒΑΪΑ

Επιβλέπουσα : Δασκαλοπούλου Ασπασία, Επίκουρη Καθηγήτρια

Διμελής εξεταστική επιτροπή :

Ακρίτας Αλκιβιάδης, Αναπληρωτής Καθηγητής
Δασκαλοπούλου Ασπασία, Επίκουρη Καθηγήτρια

Βόλος, Μάρτιος 2013



Department of Computer & Communication Engineering
School of Engineering of the University of Thessaly

Aco: A bio-inspired algorithm for ad-hoc and cloud networks

Design and implementation of an ant-based system simulator

ZYGOURA VAIA

Volos, March 2013

.....
Ζυγούρα Βαΐα

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών Τηλεπικοινωνιών &
Δικτύων Π.Θ

Copyright © Ζυγούρα Βαΐα

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Τις τελευταίες δεκαετίες τόσο οι βιολόγοι όσο και επιστήμονες στο χώρο της πληροφορικής και των επικοινωνιών μελετούν τα μοντέλα των σμηνών προσπαθώντας να κατανοήσουν τον τρόπο που τα μέλη τους αναπτύσσονται, αντιδρούν και επιτυγχάνουν στόχους μέσα στην φύση. Η μελέτη αυτή μπορεί να τους οδηγήσει στο να θεμελιώσουν και να χρησιμοποιήσουν βίο-εμπνευσμένους αλγόριθμους σε προβλήματα βελτιστοποίησης, κατασκευής προτύπων και αυτό-οργάνωσης συστημάτων, με εξαιρετική επιτυχία.

Ένας τέτοιος αλγόριθμος δημιουργήθηκε στην προσπάθεια μίμησης της συμπεριφοράς που αναπτύσσουν οι αποικίες μυρμηγκιών και ονομάστηκε Ant Colony Optimization (ACO). Το πρώτο σύστημα που υλοποίησε τον ACO εισήχθη από τον Marco Dorigo. Ο ACO είναι ένας *μεθευριστικός* αλγόριθμος γενικού σκοπού, ο οποίος μπορεί να χρησιμοποιηθεί για την επίλυση NP-complete προβλημάτων και είχε πολλές εξελίξεις. Ο αλγόριθμος ACO αποτελεί ιδανική λύση για δίκτυα που χαρακτηρίζονται από αυτονομία, λόγω της αυτοργάνωσης που παρουσιάζει. Ο ACO αποτέλεσε πηγή έμπνευσης για την δημιουργία πρωτοκόλλων δρομολόγησης σε ad-hoc δίκτυα ενώ αποτελεί ενδιαφέρον κομμάτι έρευνας για τα Clouds.

Η εργασία αυτή περιγράφει την τρέχουσα κατάσταση της έρευνας στον τομέα της *ευφυΐας του σμήνους* (Swarm Intelligence) που βασίζεται στις αρχές του στιγματισμού για την επίλυση σύνθετων προβλημάτων χρησιμοποιώντας απλούς πράκτορες (Κεφάλαιο 1ο). Γίνεται ανάλυση του αλγορίθμου ACO (Κεφάλαιο 2ο) ενώ στη συνέχεια περιγράφεται εφαρμογή που υλοποιήθηκε και προσομοιώνει την κίνηση των μυρμηγκιών. Κομμάτι της προσομοίωσης είναι και η επίδειξη της χρήσης συγκεκριμένου αλγορίθμου για την επαναφορά της ισορροπίας ενός δικτύου σε ένα ασφαλές επίπεδο, μετά από επίθεση που δέχτηκε (Κεφάλαιο 3ο). Τέλος αναλύονται οι εφαρμογές και οι επεκτάσεις του αλγορίθμου ACO σε ad-hoc δίκτυα (Κεφάλαιο 4ο) και Clouds (Κεφάλαιο 5ο).

Abstract

During the past few decades, biologists and computer scientists alike, have been studying swarm models in order to understand the way their members grow, react and achieve goals in nature. This study may lead them to establish and use bio-inspired algorithms in optimization problems (e.g. routing in wireless networks), design of prototypes (e.g. in transportation systems) and self-organization of systems (e.g. swarm robots) with great success.

One such algorithm was made in an attempt to mime the behavior that is developed by ant colonies and which was named Ant Colony Optimization (ACO). The first system that ACO developed was introduced by Marco Dorigo. ACO is a metaheuristic algorithm of general purpose that can be used for the solution of NP-complete problems and it has had many developments. ACO algorithm is an ideal solution for networks that are characterized by autonomy due to its self-organization behavior. ACO has been a source of inspiration for the development of routing protocols in ad-hoc networks while it constitutes an interesting part of research on Clouds.

This thesis describes the current situation of research in the field of Swarm Intelligence which is based in the principles of *stigmergy* for the solution of complex problems by using simple agents (Chapter 1). There is an analysis of the ACO algorithm (Chapter 2) while afterward an application which was developed and simulates the movement of ants is described. Part of the simulation is also the display of a specific algorithm for the retrieval of a network balance to a safe level after the attack it underwent (Chapter 3). Finally, the applications and extensions of ACO algorithm in ad-hoc networks (Chapter 4) and Clouds (Chapter 5) are analyzed.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω την καθηγήτριά μου Δασκαλοπούλου Ασπασία, για την αμέριστη βοήθεια και ελευθερία που μου πρόσφερε στην ανάπτυξη ενός συναρπαστικού θέματος. Η εκπόνηση της διπλωματικής εργασίας δεν υπήρξε ένα διεκπεραιωτικό καθήκον, αλλά μια ευχάριστη ενασχόληση. Οδήγησε δε, σε βαθιά κατανόηση ενός πολύπλοκου αλλά άκρως ενδιαφέροντος θέματος που θα μπορούσε να αποτελέσει αφετηρία πιθανής περεταίρω έρευνας από μέρους μου.

Ένα μεγάλο ευχαριστώ οφείλω στην οικογένεια μου, που υπήρξε αρωγός σε όλη αυτή την προσπάθεια ολοκλήρωσης των σπουδών μου. Εύχομαι να είναι πάντα υγιείς και δυνατοί. Ευχαριστώ επίσης την Αγγελική και τον Φαίδωνα για την ώθηση που έδωσαν στα όνειρά μου καθώς και όλους τους φίλους μου που κάθε ένας με τον δικό του ξεχωριστό τρόπο προσέφερε την βοήθειά του.

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	9
1. Ευφυΐα του Σμήνους	12
1.1 Εισαγωγή.....	12
1.2 Αυτο - οργάνωση.....	14
1.2.1 Θετική ανάδραση	15
1.2.2 Αρνητική ανάδραση.....	16
1.2.3 Ενίσχυση των τυχαίων διακυμάνσεων	16
1.2.4 Αλληλεπίδραση μεταξύ των συνιστωσών του συστήματος	17
1.2.5 Επιπλέον χαρακτηριστικά συστημάτων αυτό-οργάνωσης.....	17
1.3 Στιγμεργία	18
1.4 Εφαρμογές αυτό – οργάνωσης	21
1.5 Πολύπλοκα συστήματα (complex systems).....	22
1.5.1 Ιδιότητες των Complex Network.....	22
1.6 Δημιουργία πολυπρακτορικών συστημάτων τύπου <i>Swarm</i>	23
1.6.1 Ταξινόμηση στιγμεργετικών συμπεριφορών	25
1.7 Θετικά και Αρνητικά των <i>SI</i> συστημάτων.....	26
1.7.1 Θετικά χαρακτηριστικά.....	26
1.7.2 Αρνητικά χαρακτηριστικά.....	27
2. Ant Colony Optimization (ACO)	30
2.1 Εισαγωγή.....	30
2.2 <i>NP-hard</i> προβλήματα	30
2.3 Πρόβλημα του Περιπλανώμενου Πωλητή - <i>TSP</i>	32
2.4 Μεθευρετικοί (meta-heuristics) αλγόριθμοι	32
2.4.1 Αλγόριθμοι τοπικής αναζήτησης	34
2.5 Πραγματικά μυρμήγκια.....	38
2.5.1 Αναζήτηση τροφής.....	38
2.5.2 Αποφυγή Εμποδίων	39
2.6 Ψηφιακά μυρμήγκια	41
2.7 Αλγόριθμοι <i>ACO</i>	42
2.7.1 Δομή αλγορίθμου	43
2.7.2 Εξέλιξη και Παραλλαγές του αλγορίθμου <i>ACO</i>	46
2.8 Αποτελέσματα - Συμπεράσματα	53
2.8.1 Σύγκριση των αλγορίθμων <i>ACO</i>	53
2.8.2 Σύγκριση με άλλους γενετικούς αλγορίθμους	54
2.8.3 Συμπεράσματα	55
3 Κατασκευή προσομοιωτή αποικίας μυρμηγκιών	56
3.1 Εισαγωγή.....	56
3.2 Γενική Περιγραφή συστήματος.....	56
3.3 Ανάλυση γραφικού περιβάλλοντος	59
3.3 <i>Ant Colony Simulation</i>	64
3.3.1 Εύρεση βέλτιστου μονοπατιού	64

3.3.2 Προσαρμοστικότητα Ant Colony Simulator	66
3.4 <i>Network Security Simulation</i>	68
3.4.1 Στιγμιότυπα εκτέλεσης Network Simulator	69
3.4.2 Προσαρμοστικότητα Network Simulator	70
3.5 <i>Σενάρια εκτέλεσης</i>	70
3.5.1 Το πείραμα της διπλής γέφυρας	70
3.5.2 Παράδειγμα λαβύρινθου	73
3.6 <i>Πειραματισμός με ιδιότητες του συστήματος</i>	74
3.6.1 Pheromone	75
3.6.2 Evaporation	76
3.6.3 DecreaseNP	76
3.6.3 Follow Strict	77
3.7 <i>Παρόμοιες εφαρμογές προσομοίωσης</i>	78
3.7.1 NetLogo	78
3.7.2 Repast	78
3.7.3 Mason	79
3.8 <i>Μειονεκτήματα και μελλοντική εξέλιξη της εφαρμογής</i>	80
4 Αλγόριθμος ACO σε ad-hoc δίκτυα	81
4.1 <i>Εισαγωγή</i>	81
4.2 <i>Κινητά ad hoc δίκτυα (MANET)</i>	82
4.2.1 Χαρακτηριστικά των δικτύων MANET	83
4.2.2 Εφαρμογές των δικτύων MANET	84
4.3 <i>Πρωτόκολλα δρομολόγησης σε δίκτυα MANET</i>	85
4.3.1 Δυναμικά πρωτόκολλα δρομολόγησης	85
4.3.2 Αντιδραστικά πρωτόκολλα δρομολόγησης	85
4.3.3 Υβριδικά πρωτόκολλα δρομολόγησης	86
4.3.4 Ιεραρχικά πρωτόκολλα δρομολόγησης	86
4.3.5 Γεωγραφικά πρωτόκολλα δρομολόγησης	86
4.4 <i>Υπάρχοντα πρωτόκολλα δρομολόγησης</i>	88
4.5 <i>Πρωτόκολλα δρομολόγησης βασισμένα στις αποικίες μυρμηγκιών</i>	89
4.5.1 Πρωτόκολλο δρομολόγησης AntHocNet	89
4.5.2 Πρωτόκολλο δρομολόγησης Accelerated Ants	89
4.5.3 Πρωτόκολλο δρομολόγησης Source Update	91
4.5.4 Πρωτόκολλο δρομολόγησης Random Walk	92
4.5.5 Πρωτόκολλο δρομολόγησης PERA	94
4.5.6 Πρωτόκολλο δρομολόγησης ARA	94
4.5.7 Πρωτόκολλο δρομολόγησης AntAODV	96
5 Αλγόριθμος ACO σε Clouds	97
5.1 <i>Εισαγωγή</i>	97
5.2 <i>Τύποι του cloud computing</i>	98
5.2.1 SaaS (software-as-a-service)	98
5.2.2 PaaS (platform-as-a-service)	99
5.2.3 IaaS (infrastructure-as-a-service)	99
5.3 <i>Χαρακτηριστικά του Cloud Computing</i>	99
5.4 <i>Χρήση ACO αλγορίθμων σε Clouds</i>	101
5.4.1 Χρήση του Aco για load balancing στα Clouds	101

5.4.2 Χρήση του Aco για τοποθέτηση φόρτου εργασίας σε Clouds με περιορισμένη κατανάλωση ενέργειας.....	103
5.4.3 Στρατηγική χρονοδρομολόγησης βάση ACO για χρήση πόρων σε Clouds.....	104
6 Συμπεράσματα - Επίλογος	106
Βιβλιογραφία	107
Αναφορές – Πηγές από το internet.....	110

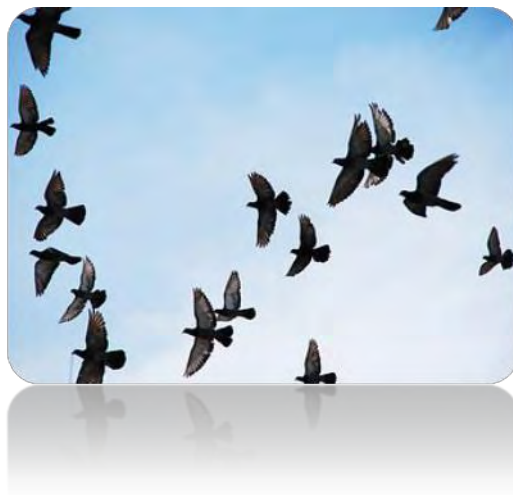
Κεφάλαιο 1ο

1. Ευφυΐα του Σμήνους

1.1 Εισαγωγή

Τα κοινωνικά έντομα και ζώα πάντοτε εντυπωσίαζαν τους ανθρώπους, επιστήμονες και μη, με τη δομή και την οργάνωσή τους. Ένα σμήνος πουλιών που σχίζει τον αέρα, ένα μελίσσι που εκτελεί το χορό των μελισσών στην αναζήτησή της τροφής τους, κοπάδια ψαριών που κινούνται, στρίβουν, αποφεύγουν τους κινδύνους όλα μαζί. Ονομάζουμε αυτή την συλλογική κίνηση, **συμπεριφορά σμήνους** (Swarm Behavior).

Τόσο οι βιολόγοι όσο και επιστήμονες στο χώρο της πληροφορικής και των επικοινωνιών μελετούν τα μοντέλα των σμηνών προσπαθώντας να κατανοήσουν τον τρόπο που τα μέλη τους αναπτύσσονται, αντιδρούν και επιτυγχάνουν στόχους μέσα στην φύση. Η μελέτη αυτή μπορεί να τους οδηγήσει στο να θεμελιώσουν και να χρησιμοποιήσουν βίο- εμπνευσμένους (bio-inspired) αλγορίθμους σε προβλήματα βελτιστοποίησης (π.χ. δρομολόγησης σε ασύρματα δίκτυα), κατασκευής προτύπων (π.χ. σε συστήματα μεταφοράς), και αυτό-οργάνωσης (π.χ. σμήνη ρομπότ) με εξαιρετική επιτυχία.



Εικόνα 1.1 Σμήνος πουλιών

Ένα σμήνος λειτουργεί χωρίς επίβλεψη. Έτσι για παράδειγμα ένα χελιδόني που θέλει να μεταναστεύσει με άλλα χελιδόνια, το μόνο που θα πρέπει να κάνει είναι να συντονίσει τις κινήσεις των φτερών του με τις κινήσεις των φτερών των υπόλοιπων μελών της ομάδας.

Κανένα πουλί δεν παίρνει εντολές από κάποιο πουλί “αρχηγό” και οι μόνες κινήσεις που κάνει είναι να μετακινείται μπροστά, στο κέντρο ή πίσω χωρίς να ξεφεύγει από τον σχηματισμό σμήνους (Εικ.1.1). Η συμπεριφορά σμήνους βοηθάει

τα πουλιά να αναζητούν την τροφή τους με επιτυχία, να προστατεύονται από τα αρπακτικά πουλιά κ.α.

Στη πραγματικότητα, η ομαδική τους εργασία αυτοργανώνεται και ο συντονισμός αναδύεται από τις αλληλεπιδράσεις ανάμεσα στα άτομα του σμήνους. Αν και αυτές οι αλληλεπιδράσεις μπορεί να είναι πρωτόγονες και απλοϊκές, συνολικά οδηγούν σε μια αποτελεσματική λύση στα προβλήματα που αντιμετωπίζουν τα σμήνη. Η συλλογική συμπεριφορά που επιδεικνύεται από μια ομάδα κοινωνικών ζώων αποκαλείται *ευφυΐα του σμήνους* - Swarm Intelligence (SI). Διάφορες συμπεριφορές που επιδεικνύουν τα σμήνη παρουσιάζονται στον παρακάτω πίνακα (πίνακας1).

Συμπεριφορά σμήνους	Οντότητες
Pattern Generation	Bacteria, Slime mold
Path Formation	Ants
Nest Sorting	Ants
Cooperative Transport	Ants
Food Source Selection	Ants, Bees
Thermoregulation	Bees
Task Allocation	Wasps
Hive Construction	Bees, Wasps, Hornets, Termites
Synchronization	Fire Flies
Feeding Aggregation	Bark Beetles
Web Construction	Spiders
Schooling	Fishes
Flocking	Birds

Πίνακας 1.1 Συμπεριφορά σμήνους σε διάφορες οντότητες

1.2 Αυτο - οργάνωση

Εξαιρετικό ενδιαφέρον, παρουσιάζει η μελέτη της συμπεριφοράς που αναπτύσσουν τα μυρμήγκια στις αποικίες τους. Ζούνε εδώ και 100 εκατομμύρια χρόνια έχοντας αποικήσει κάθε μέρος στον πλανήτη μας ενώ ο πληθυσμός τους υπολογίζεται γύρω στα 10^{16} άτομα . Το μυρμήγκι από μόνο του εμφανίζει αρκετά περίπλοκη δομή χρησιμοποιώντας πληθώρα αισθητηρίων οργάνων ώστε να συγκεντρώνει πληροφορίες από το περιβάλλον του και να λαμβάνει τις ανάλογες αποφάσεις. Για παράδειγμα, είναι αξιοσημείωτο πως τα μυρμήγκια *Cataglyphis fortis* καταφέρνουν να εντοπίσουν τον προορισμό τους (τοποθεσία με φαγητό) χρησιμοποιώντας οπτικά δεδομένα από τη μορφολογία του περιβάλλοντος, τον ήλιο ως πυξίδα, τη μυρωδιά του στόχου, την περιορισμένη μνήμη τους αλλά και ένα αθροιστή που διαθέτουν προκειμένου να υπολογίζουν αποστάσεις .

Παραπάνω είδαμε μερικές από τις ατομικές ικανότητες που έχει κάθε έντομο όταν εκτελεί τις εργασίες του. Οι εργασίες αυτές χαρακτηρίζονται από μεγάλη ευελιξία αλλά και προσαρμοστικότητα (*flexibility*). Δηλαδή τα έντομα έχουν την ικανότητα να προσαρμόζονται στις εκάστοτε συνθήκες του περιβάλλοντος και επιπλέον διαθέτουν μεγάλη ανθεκτικότητα (*robustness*) μιας και μπορούν να φέρουν εις πέρας τον στόχο τους, ακόμα και αν κάποια από τα μέλη της κοινότητας, αποτύχουν.

Βέβαια αυτές οι ικανότητες των εντόμων μεμονωμένα, δεν είναι δυνατό να εμφανίσουν ένα συνολικό αρμονικά δομημένο αποτέλεσμα, όπως το κτίσιμο των φωλιών στους τερμίτες ή την εύρεση της συντομότερης διαδρομής μεταξύ δύο σημείων στα μυρμήγκια, και μάλιστα χωρίς να υπάρχει κάποιο έντομο που να κατέχει τον συντονιστικό ρόλο.

Το μυστικό της «επιτυχημένης» διεκπεραίωσης συλλογικών εργασιών και μάλιστα με βέλτιστο τρόπο βρίσκεται στην **αυτο-οργάνωση** (*self-organization* SO) των εντόμων. Στα SO μοντέλα γίνεται η βασική θεώρηση ότι το κάθε άτομο-έντομο είναι ένας απλός «πράκτορας» (*agent*) που μπορεί να διεκπεραιώσει μόνο απλές λειτουργίες. Η θεώρηση αυτή δεν λαμβάνει υπόψη την περίπλοκη δομή του μεμονωμένου εντόμου.

Για παράδειγμα, στις αποικίες των μυρμηγκιών το βασικό χαρακτηριστικό που λαμβάνουν υπόψη τα SO μοντέλα είναι ότι τα μυρμήγκια μπορούν να ακολουθήσουν με κάποια πιθανότητα ένα ίχνος φερομόνης και να το ενισχύσουν χωρίς να ενδιαφέρει πως ανιχνεύεται το ίχνος αυτό, πως λαμβάνεται η απόφαση από το μυρμήγκι να το ακολουθήσει ή όχι και τέλος με ποιον τρόπο ενισχύει το ήδη υπάρχον ίχνος φερομόνης. Τα SO μοντέλα που έχουν αναπτυχθεί με βάση την παραπάνω θεώρηση δείχνουν ότι είναι δυνατόν να εμφανιστούν περίπλοκες συλλογικές συμπεριφορές σε μια αποικία εντόμων.

Αναλυτικότερα, η **αυτο-οργάνωση** είναι ένα σύνολο δυναμικών μηχανισμών με τους οποίους σχηματίζονται δομές σε ένα σύστημα από αλληλεπιδράσεις των συνιστωσών του. Οι αλληλεπιδράσεις αυτές στηρίζονται καθαρά σε τοπικές πληροφορίες χωρίς να λαμβάνουν υπόψη τους τη συνολική εικόνα.

Η αυτο-οργάνωση χαρακτηρίζεται από 4 βασικά στοιχεία.

1. τη **θετική ανάδραση** (*positive feedback*)
2. την **αρνητική ανάδραση** (*negative feedback*)
3. την **ενίσχυση των τυχαίων διακυμάνσεων** (*amplification of random fluctuations*) και
4. την **αλληλεπίδραση μεταξύ των συνιστωσών του συστήματος**

1.2.1 Θετική ανάδραση

Η **θετική ανάδραση** είναι ένα σύνολο κανόνων το οποίο είναι υπεύθυνο για τη δημιουργία των βασικών δομών. Κατά τη συγκεκριμένη διεργασία ένα έντομο αντιδρά στην ενεργοποίηση από ένα ερέθισμα (πχ εύρεση τροφής ή εντοπισμός ίχνους φερομόνης) φροντίζοντας για την προσέλκυση και άλλων εντόμων ή την ενίσχυση του ερεθίσματος.

Γενικά, η θετική ανάδραση περιλαμβάνει δυο μηχανισμούς τη **στρατολόγηση** (*recruitment*) και την **ενίσχυση** (*reinforcement*). Για την καλύτερη κατανόηση των δύο αυτών μηχανισμών παραθέτουμε παρακάτω αντιπροσωπευτικά παραδείγματα για κάθε μία από τις περιπτώσεις.

- **Παράδειγμα στρατολόγησης** : Έστω ότι ένα μυρμήγκι εντοπίζει μια τοποθεσία με τροφή. Κατά την επιστροφή του στη φωλιά εναποθέτει φερομόνη. Το ίχνος αυτό της φερομόνης είναι ανιχνεύσιμο από τα υπόλοιπα μυρμήγκια της αποικίας τα οποία είναι πιθανό να προτιμήσουν να ακολουθήσουν την ίδια διαδρομή με μεγαλύτερη πιθανότητα από μια άλλη νέα-τυχαία διαδρομή. Με αυτόν τον τρόπο ένα μυρμήγκι καταφέρνει να «στρατολογήσει» και άλλα μυρμήγκια με σκοπό να εκμεταλλευθούν την ίδια **πηγή τροφής** (*food source*).
- **Παράδειγμα ενίσχυσης** : Έστω ένα μυρμήγκι εντοπίζει ένα ίχνος φερομόνης. Με κάποια πιθανότητα αποφασίζει να το ακολουθήσει. Αν το ακολουθήσει τότε ενισχύει το ίχνος αυτό εναποθέτοντας και την δική του φερομόνη. Με τον τρόπο αυτό η διαδρομή που ενισχύεται με επιπλέον φερομόνη γίνεται πιο ελκυστική για τα επόμενα μυρμήγκια.

1.2.2 Αρνητική ανάδραση

Η αρνητική ανάδραση αναιρεί τα αποτελέσματα της θετικής ανάδρασης και βοηθάει στη σταθεροποίηση σε κάποια συγκεκριμένη δομή. Η αρνητική ανάδραση μπορεί να μοντελοποιεί:

- την εγκατάλειψη κάποιου μονοπατιού από τα μυρμήγκια. Για παράδειγμα, έχει παρατηρηθεί ότι η φερομόνη που εναποθέτουν τα μυρμήγκια στη διαδρομή τους εξατμίζεται με κάποιο ρυθμό. Αυτό έχει ως αποτέλεσμα την εγκατάλειψη των διαδρομών που για διάφορους εξωγενείς λόγους δεν χρησιμοποιούνται συχνά
- την εξάντληση κάποιας πηγής τροφής. Για παράδειγμα, οι μέλισσες σταματούν να εκτελούν το χορό των μελισσών για μια τοποθεσία στην οποία η τροφή έχει εξαντληθεί και επομένως δεν στρατολογούνται άλλες μέλισσες για εκμετάλλευση της περιοχής αυτής.
- τον ανταγωνισμό μεταξύ δύο διαφορετικών πηγών τροφής. Για παράδειγμα, όταν δύο μέλισσες έχουν εντοπίσει δύο διαφορετικές πηγές τροφής, εκτελούν το χορό των μελισσών στη φωλιά τους. Η μέλισσα με την καλύτερη ποιοτικά και ποσοτικά πηγή τροφής «στρατολογεί» και τα περισσότερα μέλη της αποικίας απομακρύνοντας το ενδιαφέρον από την άλλη περιοχή.

1.2.3 Ενίσχυση των τυχαίων διακυμάνσεων

Η ενίσχυση των τυχαίων διακυμάνσεων είναι σημαντικότερη διαδικασία καθώς μπορεί να οδηγήσει στη δημιουργία νέων βελτιωμένων λύσεων ή και ακόμα και της συνολικά βέλτιστης λύσης ξεφεύγοντας ενδεχόμενα από τοπικά βέλτιστες συμπεριφορές. Οι τυχαίοι παράγοντες που υπεισέρχονται στα πραγματικά βιολογικά συστήματα είναι πολλοί και καθοριστικοί. Για παράδειγμα, είναι πολύ πιθανό να χαθεί ένα μυρμήγκι κατά την αναζήτηση και συγκομιδή τροφής και τελικά να ανακαλύψει μια καινούργια καλύτερη πηγή τροφής «στρατολογώντας» και άλλα μέλη της αποικίας του για την εκμετάλλευση της πηγής αυτής.

1.2.4 Αλληλεπίδραση μεταξύ των συνιστωσών του συστήματος

Κάθε μοντέλο αυτό-οργάνωσης βασίζεται στην **αλληλεπίδραση μεταξύ των ατόμων του**. Κάθε άτομο-έντομο πρέπει να είναι σε θέση να δημιουργήσει κάποια σταθερή δομή (πχ ένα μονοπάτι φερομόνης) αλλά και να ανιχνεύσει της δομές που δημιούργησαν άλλα άτομα της κοινότητάς του. Για να έχει αποτέλεσμα η αλληλεπίδραση θα πρέπει να υπάρχει ένας ελάχιστος πληθυσμός ατόμων-εντόμων, διαφορετικά η εμφάνιση οργανωμένων δομών μέσω της συλλογικής δράσης είναι αδύνατη διότι υπάρχουν τα φαινόμενα αρνητικής ανάδρασης

1.2.5 Επιπλέον χαρακτηριστικά συστημάτων αυτό-οργάνωσης

Τέλος, μερικά από τα χαρακτηριστικά που παρατηρούνται σε συστήματα αυτό-οργάνωσης είναι:

- Η εμφάνιση συγκεκριμένων δομών σε αρχικώς ομοιογενή μέσα. Για παράδειγμα, η δημιουργία μονοπατιών φερομόνης από τα μυρμήγκια ή η κατασκευή φωλιάς με εναπόθεση σφαιριδίων χρώματος από τους τερμίτες. Ως **μονοπάτι φερομόνης** (*pheromone trail*) ορίζεται κάθε διαδρομή στην οποία έχει εναποτεθεί κάποια ποσότητα φερομόνης από ένα ή/και περισσότερα μυρμήγκια. Έχει ήδη αναφερθεί ότι η φερομόνη εξατμίζεται με κάποιο ρυθμό. Συνεπώς, αν ένα μυρμήγκι δημιουργήσει μια νέα επιτυχημένη διαδρομή αλλά ο πληθυσμός των μυρμηγκιών δεν είναι αρκετός ώστε να ενισχύσει τη διαδρομή αυτή, τότε το νέο αυτό μονοπάτι θα εγκαταλειφθεί.
- Η εν δυνάμει εμφάνιση πολλών αποδεκτών-σταθερών λύσεων (*multistability*) ανάλογα με τις αρχικές συνθήκες. Για παράδειγμα, έστω ότι υπάρχουν δύο τοποθεσίες τροφής A και B σε ίση απόσταση από τη φωλιά των μυρμηγκιών αλλά και με την ίδια ποιότητα και ποσότητα τροφής. Η τελική αξιοποίηση μιας εκ των δύο πηγών τροφής εξαρτάται κατά κύριο λόγο από τις αρχικές τυχαίες διακυμάνσεις. Αλλά και οι δύο λύσεις, δηλαδή η συγκομιδή τροφής είτε από την τοποθεσία A είτε από την B, είναι αποδεκτές.
- Η ύπαρξη δύο καταστάσεων συμπεριφοράς των εντόμων κατά τη διεκπεραίωση μιας συλλογικής εργασίας. Για παράδειγμα, η μετάβαση από την τυχαία εναπόθεση σφαιριδίων χρώματος στην οργανωμένη κατά το κτίσιμο φωλιάς από τους τερμίτες.

1.3 Στιγμεργία

Πολλές φορές τα κοινωνικά έντομα προκειμένου να επιτύχουν τις συλλογικές εργασίες τους πρέπει να επικοινωνήσουν είτε άμεσα είτε έμμεσα. Η άμεση επικοινωνία βασίζεται στην οπτική ή χημική επαφή των εντόμων, στις ανταλλαγές τροφής ή υγρών μεταξύ των εντόμων κτλ.

Αντίθετα, η έμμεση επικοινωνία βασίζεται στις μεταβολές του περιβάλλοντος από τα έντομα και την αντίγνευση των μεταβολών αυτών από τα ίδια ή άλλα έντομα της αποικίας. Ένα χαρακτηριστικό παράδειγμα τέτοιας διεργασίας είναι η εναπόθεση φερομόνης από τα μυρμήγκια στο έδαφος κατά τη μεταφορά τροφής στη φωλιά. Η φερομόνη αυτή είναι εντοπίσιμη και από τα υπόλοιπα μέλη της αποικίας τα οποία δρουν ανάλογα.

Η παραπάνω διαδικασία έμμεσης επικοινωνίας των εντόμων χαρακτηρίζεται από τον όρο **στιγμεργία** (*stigmergy*) ο οποίος προέρχεται από τις ελληνικές λέξεις **στίγμα** (*stigma*) και **έργο** (*ergo*) και εισήχθη για πρώτη φορά από τον Grassé στο έργο του για τους τερμίτες *Bellicositermes Natalensis* και *Cubitermes*. Βασικά, στο έργο αυτό είχε ορίσει τον όρο στιγμεργία (*stigmergy*) ως

“τόνωση των εργατών βάση της απόδοσης που έχουν επιτύχει”

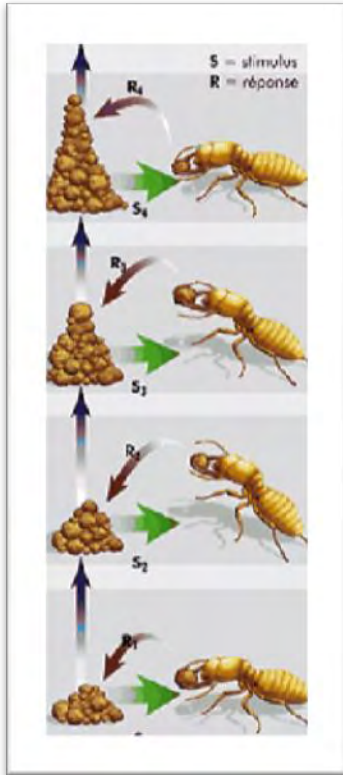
Ο Grassé παρατήρησε ότι τα έντομα διεγείρονται εκτελώντας μια γενετικά προκαθορισμένη αντίδραση όταν δεχτούν ένα σημαντικό ερέθισμα. Η αντίδραση αυτή μπορεί να αποτελεί ένα νέο σημαντικό ερέθισμα τόσο για το ίδιο το έντομο που την ενεργοποίησε αρχικά όσο και για τα υπόλοιπα μέλη της αποικίας του. Πιο συγκεκριμένα, ο Grassé παρατήρησε τη διαδικασία δημιουργίας φωλιάς στους τερμίτες *Bellicositermes Natalensis* και *Cubitermes* (**Εικ. 1.2**).



Εικόνα 1.2 Φωλιά Τερμιτών

Η διαδικασία αποτελεί χαρακτηριστικό παράδειγμα στιγμεργίας και έχει ως εξής:

Οι τερμίτες εργάτες, χρησιμοποιούν μικρά σφαιρικά κομμάτια χώματος τα οποία έχουν φροντίσει να τα εμποτίσουν με φερομόνη (χημική ουσία που παράγουν οι ίδιοι) προκειμένου να χτίσουν τους πυλώνες της φωλιάς τους. Αυτό γίνεται σε δύο φάσεις :



1. αρχικά και εντελώς τυχαία οι τερμίτες εργάτες εναποθέτουν μικρά κομμάτια χώματος το ένα πάνω από το άλλο. Αυτό γίνεται μέχρι ο σωρός που δημιουργείτε από τα σφαιρίδια να φτάσει σε ένα κρίσιμο μέγεθος. Σε αυτή την φάση δεν υπάρχει κανένας απολύτως συντονισμός (Εικ. 1.3) .

2. στην συνέχεια, ακολουθεί μια φάση όπου υπάρχει πλήρης συντονισμός μεταξύ των ομάδων εργασίας. Η ύπαρξη ενός αρχικού λόφου χωμάτινων σβόλων, με διάχυτη φερομόνη πάνω του, διεγείρει τους υπόλοιπους εργάτες να συσσωρεύουν όλο και περισσότερο υλικό μέσω του μηχανισμού θετικής ανάδρασης. Όσο μεγαλύτερος είναι ο σωρός τόσο πιο ελκυστικός γίνεται για την εναπόθεση νέων σφαιριδίων σε αυτόν. Έτσι θα σχηματιστεί μια στήλη ενώ αργότερα, θα σχηματιστούν τα τόξα που ενώνουν τις στήλες μεταξύ τους και τελικά θα δημιουργηθεί η φωλιά. Οι μικρότεροι λόφοι γίνονται λιγότερο ελκυστικοί με την πάροδο του χρόνου μιας και η φερομόνη εξατμίζεται.

Εικόνα 1.3 Κατασκευή φωλιάς

Σε αυτό το παράδειγμα φαίνονται και τα τέσσερα χαρακτηριστικά της αυτό-οργάνωσης όπως αναφέρθηκαν στο 1.1 .Συγκεκριμένα :

- θετική ανάδραση → σαν φαινόμενο χιονοστιβάδας ο ένας τερμίτης παρασύρει τον άλλον για να εναποθέσει σε συγκεκριμένο σημείο το σφαιρίδιο με το χώμα
- αρνητική ανάδραση → η φερομόνη εξατμίζεται σιγά σιγά
- ενίσχυση των τυχαίων διακυμάνσεων → πυλώνες θα μπορούσαν να δημιουργηθούν σε οποιοδήποτε σημείο
- πολλαπλές αλληλεπιδράσεις → μέσω του περιβάλλοντος

και αυτό είναι πολύ λογικό μιας και

“στιγμεργία είναι η στρατηγική που ακολουθεί η ίδια η φύση προκειμένου τα έντομα να αυτό-οργανώνονται” .

Άλλο ένα χαρακτηριστικό παράδειγμα της δομής και λειτουργίας των εντόμων είναι ο τρόπος με τον οποίο πετούν και επικοινωνούν οι μέλισσες. Οι μέλισσες καθορίζουν την ταχύτητά τους και μετρούν την απόσταση που έχουν διανύσει ανάλογα με την «οπτική ροή» των πληροφοριών στον αμφιβληστροειδή: π.χ. πετώντας χαμηλά και διανύοντας μικρή απόσταση δημιουργείται η ίδια ροή πληροφοριών με πετώντας ψηλά και διανύοντας μεγάλη απόσταση. Χρησιμοποιώντας τον ίδιο μηχανισμό και συνδυάζοντας το ρυθμό καθόδου τους με την πρόσω ταχύτητά τους καταφέρνουν να επιτυγχάνουν πάντα ομαλές προσγειώσεις σε οριζόντια επίπεδα χωρίς να έχουν σαφή γνώση τόσο του ύψους τους όσο και της ταχύτητάς τους.

Τέλος, είναι γνωστό ότι η επικοινωνία μεταξύ των μελισσών επιτυγχάνεται μέσω του περίφημου χορού των μελισσών (*waggle dance of honeybees*) ο οποίος κωδικοποιεί την απόσταση και τη διεύθυνση της θέσης της τροφής. Σε ένα μελίσσι, τα άτομα εξειδικεύονται σε συγκεκριμένες εργασίες, για την αποδοτικότερη λειτουργία της κυψέλης (**Εικ. 4**). Παρόλα αυτά η ανάθεση εργασιών παραμένει πολύ ευέλικτη. Όταν το φαγητό είναι δυσεύρετο όλες οι μέλισσες θα ψάξουν για αυτό ανεξαρτήτου της δουλειάς που έχει αναλάβει η κάθε μια να εκτελέσει. Οι πληθυσμοί των εξειδικευμένων ατόμων ως προς κάποια εργασία συνεχώς μεταβάλλονται ανάλογα με τις ανάγκες της κοινότητας.



Εικόνα 1.4 Μελίσσι

Οι Bonabeau, Dorigo και Theraulaz στο συμπλήρωσαν τον ορισμό του Grassé και του έδωσαν την εξής μορφή:

Στιγμεργετική (*stigmergetic*) επικοινωνία είναι κάθε έμμεση επικοινωνία η οποία πραγματοποιείται με φυσικές μεταβολές στο περιβάλλον οι οποίες είναι μόνο τοπικά προσπελάσιμες από τα επικοινωνούντα άτομα (*agents*).

Τέλος, θα πρέπει να επισημανθεί ότι η στιγμεργετική επικοινωνία συμβάλλει στην ευελιξία ή/και προσαρμοστικότητα του συστήματος στο περιβάλλον. Κάθε εξωγενής αλλαγή στις παραμέτρους του περιβάλλοντος μπορεί να ληφθεί από τα άτομα της αποικίας ως ενδογενής αλλαγή από κάποια άλλα άτομα της αποικίας και να αντιμετωπισθεί ανάλογα

1.4 Εφαρμογές αυτό – οργάνωσης

Τα αυτό-οργανωμένα συστήματα αποτελούν την νέα τάση που επιλέγουν να ακολουθήσουν οι επιστήμονες για την δημιουργία τεχνολογικών και μη, εφαρμογών. Παραδείγματος χάρη, πολλές εφαρμογές έχουν αναπτυχθεί από τον τρόπο καταμερισμού της εργασία που εφαρμόζουν τα κοινωνικά έντομα.

Έτσι, χρησιμοποιώντας τις μέλισσες ως πρότυπο, αναπτύχθηκε ένα σύστημα για το προγραμματισμό των θαλάμων βαφής σε ένα εργοστάσιο παραγωγής φορτηγών. Κατά τη λειτουργία τους, στους θαλάμους βάφονται τα φορτηγά που εξέρχονται της γραμμής παραγωγής. Όταν είναι απαραίτητο, ένα θάλαμος μπορεί να αλλάξει το χρώμα που χρησιμοποιείται για τη βαφή, αλλά αυτό συνεπάγεται καθυστέρηση στη διαδικασία και κοστίζει. Αν θεωρήσουμε τους θαλάμους ως μέλισσες που ακολουθούν τον παρακάτω κανόνα: ένα άτομο εκτελεί ένα έργο για το οποίο είναι εξειδικευμένο εκτός εάν αντιλαμβάνεται μια σημαντική ανάγκη για να πραγματοποιήσει κάποια άλλη λειτουργία. Επομένως, ένας θάλαμος με κόκκινη μπογιά θα συνεχίσει να αναλαμβάνει παραγγελίες αυτού του χρώματος έως ότου κάποια εργασία που χαρακτηρίζεται «επείγουσα» απαιτεί ένα λευκό φορτηγό και οι ουρές στους άλλους θαλάμους, ιδιαίτερα σε αυτούς που ειδικεύονται στο λευκό, είναι κατά πολύ μεγαλύτερες. Αν και ο κανόνας μοιάζει απλός, στην πράξη είναι εκπληκτικά αποτελεσματικός. Έχει εμπλέξει τους θαλάμους βαφής στο να καθορίζουν το φορτίο τους πιο αποτελεσματικά (μέσω λιγότερων αλλαγών χρώματος) από ότι ένα κεντρικά ελεγχόμενο σύστημα.

Αυτό το αυτοργανώμενο σύστημα διαθέτει τα πλεονεκτήματα της swarm intelligence: ευελιξία και ευρωστία. Η ομάδα μπορεί να προσαρμόζεται σε ένα μεταβαλλόμενο περιβάλλον. Όταν προκύψει απρόσμενα ένα πλήθος φορτηγών που πρέπει να βαφτεί μπλε, άλλοι θάλαμοι εγκαταλείπουν το χρώμα που έχουν ειδικευτεί για να προσαρμοστούν στην έντονη ζήτηση για αυτό το χρώμα. Ενώ στην περίπτωση που ένα ή περισσότερα άτομα πάψουν να “λειτουργούν” η ομάδα συνεχίζει να εκτελεί την εργασία της, πχ. όταν ένας θάλαμος τεθεί εκτός λειτουργίας, οι υπόλοιποι ανταποκρίνονται διαμοιράζοντας το επιπλέον φορτίο

Άλλο ένα παράδειγμα είναι η κατασκευή ενός πολύ ισχυρού ενισχυτή, που αποτελείται από χιλιάδες στοιχειώδεις ενισχυτές (περιορισμένων δυνατοτήτων). Οι στοιχειώδεις αυτοί ενισχυτές είναι τοπικά διασυνδεδεμένοι μεταξύ τους (δεν υπάρχει άμεση σύνδεση- αλληλεπίδραση μεταξύ οποιαδήποτε δύο ενισχυτών). Κάθε στοιχειώδης ενισχυτής μπορεί να αλληλεπιδρά με τους γειτονικούς του. Κάθε χρονική στιγμή η κατάσταση λειτουργίας του καθορίζεται από τις καταστάσεις λειτουργίας των γειτόνων του.

Δεν έχει άμεση και ακριβή γνώση για καθένα από αυτούς. Παράδειγμα η θερμοκρασία που αναπτύσσεται στη γειτονιά του ενισχυτή καθώς και τα πεδία επηρεάζουν την απόδοσή του. Ο έλεγχος τόσο πολλών υποσυστημάτων δεν είναι εφικτός κεντρικά. Οι μορφή των αλληλεπιδράσεων και ο τρόπος που καθορίζουν την αντίδραση του κάθε ενισχυτή, αποτελούν το καταναμημένο μοντέλο ελέγχου. Σε περίπτωση βλάβης κάποιων από τους στοιχειώδεις ενισχυτές υπάρχουν χιλιάδες άλλοι που λειτουργούν.

Τα συστήματα αυτά παρουσιάζουν μεγάλη ευρωστία και προσαρμοστικότητα σε έντονα μεταβαλλόμενα περιβάλλοντα, η σπουδαιότητα των ιδιοτήτων αυτών ενισχύεται από το γεγονός ότι τα επιμέρους συστήματα ακολουθούν σχετικά απλούς κανόνες. Η ιδιότητα αυτή της ευφυΐας που παρουσιάζει το συνολικό σύστημα ενώ βασίζεται σε «απλοϊκά» συστήματα, της κοινωνικής ευφυΐας, έχει αποδεδειγμένα μεγάλη αξία.

Το ερώτημα που τίθεται είναι :

ποσότητα έναντι ποιότητας ή λιτότητα έναντι πλεονασμού;

1.5 Πολύπλοκα συστήματα (complex systems)

Στην προηγούμενη ενότητα είδαμε την σημαντικότητα της αυτό – οργάνωσης. Καταλάβαμε πως μια ομάδα δεν χρειάζεται να διαθέτει ένα ισχυρό και πολύπλοκο σύστημα κεντρικού ελέγχου. Οι απλοί κανόνες που ακολουθεί το άτομο επαρκούν για την επίτευξη της επιθυμητής λειτουργίας. Οι αποικίες των κοινωνικών εντόμων αποτελούν επιτυχημένα παραδείγματα μιας ευρύτερης οικογένειας όπου εμφανίζονται τέτοιες συμπεριφορές, **τα πολύπλοκα συστήματα** (complex systems). Πολύπλοκα συστήματα αποτελούν το διαδίκτυο, οι αγορές προϊόντων, οι νευρώνες, τα συγκοινωνιακά δίκτυα κ.α. Το όνομα τους το οφείλουν στην πολυπλοκότητα της συμπεριφοράς τους, που προκαλείται από το μεγάλο πλήθος διακριτών οντοτήτων στις οποίες συνίστανται.

1.5.1 Ιδιότητες των Complex Network

Τα πολύπλοκα συστήματα μπορεί να έχουν τα ακόλουθα χαρακτηριστικά:

1. **Όρια δύσκολο να καθοριστούν:** Μπορεί να είναι δύσκολο να καθοριστούν τα όρια ενός πολύπλοκου συστήματος. Η απόφαση αυτή τελικά εξαρτάται από τον παρατηρητή
2. **Τα πολύπλοκα συστήματα μπορεί να είναι ανοιχτά (open):** Τα πολύπλοκα συστήματα είναι συνήθως ανοικτά συστήματα - δηλαδή, μπορούν να δέχονται και να δίνουν ενέργεια. Με άλλα λόγια, τα πολύπλοκα συστήματα είναι συχνά μακριά από την ενεργειακή ισορροπία, αλλά παρ' όλη αυτή τη ροή μπορεί να υπάρξει ένα σταθερό μοτίβο

3. **Τα πολύπλοκα συστήματα μπορεί να έχουν μνήμη (memory):** Η ιστορία ενός πολύπλοκου συστήματος μπορεί να είναι σημαντική. Επειδή τα πολύπλοκα συστήματα είναι δυναμικά συστήματα που αλλάζουν με την πάροδο του χρόνου, οι προηγούμενες καταστάσεις μπορεί να επηρεάζουν τις τωρινές. Πιο συγκεκριμένα, τα περίπλοκα συστήματα συχνά παρουσιάζουν υστέρηση
4. **Τα πολύπλοκα συστήματα μπορεί να είναι ένθετα (nested):** Τα στοιχεία ενός σύνθετου συστήματος μπορεί τα ίδια να είναι πολύπλοκα συστήματα. Για παράδειγμα, μια οικονομία που αποτελείται από οργανώσεις, οι οποίες αποτελούνται από άτομα, τα οποία αποτελούνται από κύτταρα, τα οποία είναι με τη σειρά τους πολύπλοκα συστήματα
5. **Δυναμικό δίκτυο πολλαπλότητας (dynamic network of multiplicity):** Όπως και οι κανόνες ζεύξης, το δυναμικό δίκτυο ενός πολύπλοκου συστήματος είναι σημαντικό. Τα μικρού κόσμου (small world) ή χωρίς κλίμακα δίκτυα που έχουν πολλές τοπικές αλληλεπιδράσεις χρησιμοποιούνται συχνά. Φυσικά πολύπλοκα συστήματα παρουσιάζουν συχνά τέτοιες τοπολογίες. Στον ανθρώπινο φλοιό του εγκεφάλου, για παράδειγμα, βλέπουμε πυκνή τοπική συνδεσιμότητα και μερικές πολύ μεγάλες προεκβολές των αξόνων μεταξύ των περιφερειών στο εσωτερικό του φλοιού και σε άλλες περιοχές του εγκεφάλου
6. **Μπορεί να παράγουν αναδυόμενα φαινόμενα (emergent phenomena):** Τα πολύπλοκα συστήματα μπορεί να εμφανίζουν αναδυόμενες συμπεριφορές, πράγμα που σημαίνει ότι, ενώ τα αποτελέσματα μπορεί να είναι ντετερμινιστικά, μπορεί να έχουν ιδιότητες που μπορούν να μελετηθούν μόνο σε υψηλότερο επίπεδο. Για παράδειγμα, οι τερμίτες σε ένα ανάχωμα έχουν φυσιολογία, βιοχημεία και βιολογική εξέλιξη που είναι σε ένα επίπεδο ανάλυσης, αλλά η κοινωνική τους συμπεριφορά και η οικοδόμηση του αναχώματος είναι μια ιδιότητα που προκύπτει από τη συλλογική τους συμπεριφορά και πρέπει να αναλυθεί σε διαφορετικό επίπεδο
7. **Οι σχέσεις είναι μη γραμμικές (non linear):** Στην πράξη, αυτό σημαίνει ότι μια μικρή διαταραχή μπορεί να προκαλέσει ένα μεγάλο αποτέλεσμα (π.χ. φαινόμενο της πεταλούδας), ένα ευθέως ανάλογο αποτέλεσμα της αιτίας, ή ακόμα και να μην έχει καμιά επιρροή. Σε γραμμικά συστήματα, το αποτέλεσμα είναι πάντα ευθέως ανάλογο της αιτίας.
8. **Οι σχέσεις περιέχουν βρόχους ανάδρασης (feedback loops):** Η θετική και η αρνητική ανάδραση υπάρχουν συχνά στα πολύπλοκα συστήματα.

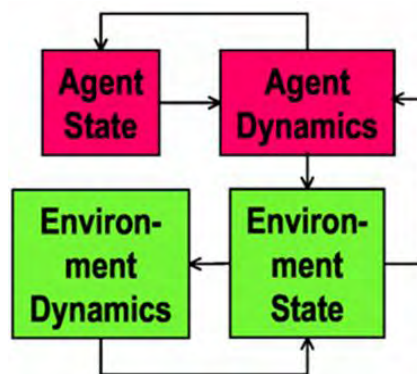
1.6 Δημιουργία πολυπρακτορικών συστημάτων τύπου Swarm

Από την σκοπιά των πολυπρακτορικών συστημάτων τώρα, θα μπορούσαμε να πούμε ότι ένα σμήνος είναι ένα σύστημα από N πράκτορες (agents) οι οποίοι συνεργάζονται προκειμένου να επιτύχουν κάποιον στόχο. Οι πράκτορες αυτοί, χρησιμοποιούν τοπικούς και απλούς κανόνες. Λειτουργούν αυτόνομα, αλλά μέσω των αλληλεπιδράσεων με άλλους πράκτορες, επιτυγχάνουν τον στόχο τους.

Η ευφυΐα ενός πολυπρακτορικού συστήματος, λοιπόν, προκύπτει από την συνεργαζόμενη ευφυΐα που επιδεικνύουν απλοί αυτόνομοι πράκτορες. Ένας αυτόνομος πράκτορας είναι ένα υποσύστημα που αλληλεπιδρά με το περιβάλλον, το οποίο πιθανόν αποτελείται και από άλλους πράκτορες, αλλά δρα σχετικά ανεξάρτητα από τους υπόλοιπους. Ένας αυτόνομος πράκτορας δεν δέχεται εντολές από κάποιον αρχηγό ή εξωτερικό παράγοντα.

Μπορούμε να κατασκευάσουμε ένα σύστημα που βασίζεται στην ευφυΐα του σμήνους ακολουθώντας τις παρακάτω αρχές που εισήγαγε ο Parunak:

1. Οι πράκτορες είναι ανεξάρτητοι και αυτόνομοι. Δεν εκτελούν λειτουργίες όπως στην περίπτωση ενός συμβατικού αντικειμενοστραφούς συστήματος.
2. Οι πράκτορες πρέπει να είναι μικροί με απλές λειτουργίες. Θα πρέπει να είναι ικανοί να αντιμετωπίσουν τον θόρυβο. Στην πραγματικότητα μάλιστα ο θόρυβος είναι ένα επιθυμητό για διάφορες μετρήσεις, χαρακτηριστικό.
3. Αποκέντρωση – δεν θα πρέπει να βασίζονται σε καθολικές πληροφορίες. Αυτό κάνει τα πράγματα πιο σταθερά.
4. Οι πράκτορες πρέπει να έχουν εξατομικευμένες συμπεριφορές – τυπικά στοχαστικές.
5. Να αφήνουν τις πληροφορίες να διαρρέουν εκτός του συστήματος
6. Οι πράκτορες πρέπει να μοιράζονται πληροφορίες – προτιμότερο δε, οι πληροφορίες αυτές να είναι τοπικές
7. Σχεδιασμός και εκτέλεση συμβαίνουν ταυτόχρονα = το σύστημα προσφέρει αντίδραση



Εικόνα 1.5 Περιβάλλον & Πράκτορας

Στο παραπάνω σχήμα (Εικόνα 1.5) συνοψίζονται οι συσχετίσεις ενός πράκτορα με το περιβάλλον του. Η κατάσταση Agent σε συνδυασμό με την κατάσταση Environment οδηγεί στην δημιουργία της δυναμικής κατάστασης του πράκτορα (δηλαδή ο πράκτορας βάση παραγόντων επιλέγει να εκτελέσει μια ενέργεια). Οι ενέργειες του Agent μέσω της τροποποίησης ή της δημιουργίας νέων σημάτων καταλήγουν σε αλλαγή της κατάστασης του Environment. Η κατάσταση

Environment χρησιμοποιείται ως είσοδος στην δυναμική κατάσταση του περιβάλλοντος. Η δυναμική κατάσταση προκαλεί αλλαγές που εμφανίζονται τελικά στην κατάσταση Environment. Αυτό που είναι σημαντικό στην παραπάνω εικόνα είναι ότι η κατάσταση του Agent είναι κρυφή. Μόνο ο ίδιος ο Agent έχει πρόσβαση σε αυτή. Αντιθέτως η κατάσταση του Environment είναι ορατή από τον Agent αλλά πρέπει να αποθηκεύεται εκ νέου εάν πρόκειται να επαναχρησιμοποιηθεί σε κάποια μεταγενέστερη χρονική στιγμή, όταν ο πράκτορας έχει (προφανώς) μετακινηθεί σε διαφορετική θέση.

1.6.1 Ταξινόμηση στιγμεργετικών συμπεριφορών

Για την εμφάνιση στιγμεργετικής συμπεριφοράς είναι απαραίτητο η συμπεριφορά του Agent να προκύπτει από αλλαγές που έγιναν κάποια στιγμή στο παρελθόν μέσα στο Environment Αυτό μπορεί να συμβεί με τρεις διαφορετικούς τρόπους:

1. Η επιλογή της αντίδρασης του Agent μπορεί να επηρεαστεί (ποιοτική επίδραση)
2. Η επιλογή της αντίδρασης δεν επηρεάζεται, αλλά μπορούν να επηρεαστούν οι παράμετροι της αντίδρασης αυτής, όπως το ακριβές σημείο, η δύναμη, η συχνότητα, η ένταση, η καθυστέρηση, κ.α. ανάλογα με την περίπτωση (ποσοτική επίδραση).
3. Μια παρελθοντική ενέργεια σε ένα σημείο δεν επηρεάζει την επιλογή ή τις παραμέτρους της επακόλουθης αντίδρασης, αλλά το αποτέλεσμα αυτής.

Η πρώτη περίπτωση περιλαμβάνει την έννοια του Grasse περί καθοδήγησης της αντίδρασης ενώ το ποιοτικό στοιχείο της επίδρασης μπορεί να ελεγχθεί εσωτερικά από κάποιο μηχανισμό κατωφλίου ο οποίος να λειτουργεί με μια ποσοτικά τροφοδοτούμενη είσοδο. Η δεύτερη περίπτωση περιλαμβάνει το στοιχείο της έντασης της δραστηριότητας. Οι δύο πρώτες περιπτώσεις αναφέρονται ως είδη ενεργητικής στιγμεργετικής συμπεριφοράς (active stigmergy) Η τρίτη περίπτωση αναφέρεται και ως παθητική στιγμεργετική συμπεριφορά (passive stigmergy), η οποία σχετίζεται με τον τρόπο δημιουργίας καθαρά φυσικών καταστάσεων – φαινομένων, στις οποίες μια δύναμη επιδρά συνεχώς αλλάζοντας το περιβάλλον με τέτοιο τρόπο έτσι ώστε αλλοιώνει το αποτέλεσμα της ίδιας στο περιβάλλον σε μελλοντικό χρόνο. Χαρακτηριστικά παραδείγματα είναι ο σχηματισμός δινών άμμου, ο σχηματισμός δέλτα των ποταμών κ.α.

Στον πίνακα 2 δίνεται μια ταξινόμια της στιγμεργετικής συμπεριφοράς σε δύο διαστάσεις. Η πρώτη διάσταση αφορά στο αν τα σημάδια στο περιβάλλον είναι αθροιστικά και ανάλογα με ένα δυναμικό πεδίο (ποσοτική στιγμεργετική συμπεριφορά – quantitative stigmergy) ή αν αποτελούν ένα σύνολο διακριτών σημάτων (ποιοτική στιγμεργετικής συμπεριφορά - qualitative stigmergy). Το άλλο επίπεδο σχετίζεται με τον τρόπο χρήσης, δηλαδή αν αποτελούν ειδικά σήματα τα οποία εκλύονται από τους Agents στο περιβάλλον (marker-based ή sign-based στιγματική συμπεριφορά) ή αν πρόκειται για καταστάσεις σηματοδότησης μιας συγκεκριμένης φάσης μιας λύσης (σηματεκτονική στιγμεργετική συμπεριφορά).

Insect Examples	Marker-Based	Sematectonic
Quantitative	Κατεύθυνση βασιζόμενη σε βαθμωτό πεδίο ενός είδους φερομόνης	Ταξινόμηση στοίβας σε αποικίες μυρμηγκιών
Qualitative	Αποφάσεις βάσει συνδυασμούς φερομονών	Κατασκευή αποικίας σφηκών

Πίνακας 1.2 Ταξινόμια στιγμεργετικής συμπεριφοράς

1.7 Θετικά και Αρνητικά των SI συστημάτων

Τα πολυπρακτορικά συστήματα βασισμένα στην κοινωνική ευφυΐα παρουσιάζουν πολλά σημαντικά πλεονεκτήματα αλλά και μειονεκτήματα. Παρακάτω θα αναφέρουμε τα πιο σημαντικά

1.7.1 Θετικά χαρακτηριστικά

Ως θετικά χαρακτηριστικά μπορούμε να αναφέρουμε τα εξής :

- Οι Agents δεν είναι προσανατολισμένοι στον στόχο. Προτιμούν να αντιδρούν και εκτελούν ενέργειες παρά να καταστρώνουν εκτενή σχέδια
- Οι Agents είναι απλοί, με περιορισμένη μνήμη και μικρή ποικιλία επιλογής στην συμπεριφορά τους
- Ο έλεγχος είναι αποκεντρωμένος. Δεν υπάρχουν καθολικές πληροφορίες στο σύστημα
- Η αποτυχία των επιμέρους Agent είναι ανεκτή. Αναδυόμενη συλλογική συμπεριφορά είναι ισχυρή και δεν επηρεάζεται από μεμονωμένες αποτυχίες
- Οι Agents μπορούν και αντιδρούν στις δυναμικές αλλαγές του περιβάλλοντος
- Δεν χρειάζονται άμεσες αλληλεπιδράσεις μεταξύ των Agents

Στον παρακάτω πίνακα (πίνακας 1.3) παρουσιάζονται σύμφωνα με τον Eric Bonabeau, μια εναλλακτική μορφή για τα θετικά στοιχεία που παρουσιάζουν τα SI συστήματα.

Ελαστικότητα (Flexible)	Η αποικία μπορεί να “απαντήσει” τόσο σε εσωτερικές διαταραχές όσο και σε εξωτερικές αλλαγές
Ανθεκτικότητα (Robust)	Οι εργασίες που εκτελούνται από τους Agents είναι ισάξιες σε περίπτωση που κάποια αποτύχει να μην προκαλέσει μεγάλο πρόβλημα
Επεκτασιμότητα (Scalable)	Από μερικούς Agents μέχρι σε εκατομμύρια
Αποκέντρωση (Decentralized)	Δεν υπάρχει κεντρικός έλεγχος/εκλεκτής στην αποικία
Αυτο-οργάνωση (Self-organized)	Τα “μονοπάτια” της λύσης προκύπτουν από τις αλληλεπιδράσεις αντί να έχουν οριστεί εξ’ αρχής

Πίνακας 1.3 Θετικά στοιχεία που παρουσιάζουν τα SI

1.7.2 Αρνητικά χαρακτηριστικά

Φυσικά παρουσιάζονται και αρκετά αρνητικά στα SI συστήματα όπως:

- a. Η συλλογική συμπεριφορά δεν μπορεί να εμφανιστεί από τις ατομικές συμπεριφορές των Agents. Αυτό συνεπάγεται πως η παρατήρηση της ατομικής συμπεριφοράς ενός Agent δεν μπορεί να μας οδηγήσει να πάρουμε μια απόφαση προκειμένου να καταλήξουμε στον στόχο μας
- b. Η μοναδική συμπεριφορά κάθε Agent μπορεί να θεωρηθεί σαν θόρυβος μιας και η επιλογή του είναι στοχαστική
- c. Ο σχεδιασμός συστημάτων τύπου Swarm είναι πολύ δύσκολος. Δεν υπάρχουν σχεδόν καθόλου αναλυτικοί μηχανισμοί για τον σχεδιασμό
- d. Παράμετροι που προσδιορίζουν τα κοινωνικά σμήνη μπορούν να έχουν πολύ σοβαρή επίδραση στην αναδυόμενη συλλογική συμπεριφορά.

Στο παρακάτω πίνακα (πίνακας 1.4) παρουσιάζονται σύμφωνα με τον Eric Bonabeau, μια εναλλακτική μορφή για τα αρνητικά στοιχεία που παρουσιάζουν τα SI συστήματα.

<i>Συμπεριφορά (Behaviour)</i>	Δυσκολία στο να προβλέψεις την συλλογική συμπεριφορά από ατομικούς ρόλους των Agents
<i>Γνώση (Knowledge)</i>	Η ερώτηση σε έναν από τα μέλη του σμήνους δεν μπορεί να δώσει απάντηση για την λειτουργία όλου του γκρουπ
<i>Εναισθησία (Sensitivity)</i>	Μικρές αλλαγές στους ατομικούς κανόνες μπορεί να οδηγήσει σε έναν εντελώς διαφορετικό συλλογικό στόχο του σμήνους ο οποίος μπορεί να παρεκκλίνει πολύ του αρχικού
<i>Ενέργειες (Actions)</i>	Ατομικές συμπεριφορές φαίνονται σαν να είναι “θόρυβος”.

Πίνακας 1.4 Αρνητικά στοιχεία που παρουσιάζουν τα SI

Από τα παραπάνω χαρακτηριστικά γνωρίσματα εμφανίζονται δυο βασικά ερωτήματα σχετικά με τον επιτυχή σχεδιασμό ενός συστήματος βασιζόμενο στη θεωρία της νοημοσύνης των σμηνών :

1. Πως καθορίζεται η ατομική συμπεριφορά και οι αλληλεπιδράσεις έτσι ώστε να επιτευχθούν τα απαραίτητα αναδυόμενα πρότυπα ;
2. Πώς σχεδιάζεται η ανάδυση αυτή;

Απάντηση στο πρώτο ερώτημα αποτελεί ο συνδυασμός εξομοίωσης και σχεδιασμού με χρήση εξελικτικού υπολογισμού. Η ιδέα της εξομοίωσης μέσω διαμεσολαβητών είναι ουσιαστικά η δημιουργία διαμεσολαβητών που λειτουργούν εφαρμόζοντας ένα σύνολο απλών κανόνων και στη συνέχεια εξομοίωση της λειτουργίας του συστήματος για ένα συγκεκριμένο χρονικό διάστημα έως ότου οι αναδυόμενες δυναμικές (αν αυτές υπάρχουν) κατοχυρωθούν.

Η αποτίμηση των αναδυόμενων δυναμικών (διαμέσω παρατηρητή ή κάποιου αυτοματοποιημένου μηχανισμού) μπορεί να χρησιμοποιηθεί για την καθοδήγηση μιας διαδικασίας μάθησης (συχνά προέρχεται από τη θεωρία του εξελικτικού υπολογισμού) η οποία αναθεωρεί τους κανόνες που χρησιμοποιήθηκαν στο συγκεκριμένο βήμα εξομοίωσης. Η επαναληπτική αυτή διαδικασία εξομοίωσης χρησιμοποιήθηκε για την υλοποίηση συμπεριφορών των swarm robots στο ερευνητικό έργο Swarm-Bots.

Απάντηση στο δεύτερο ερώτημα δίνεται μέσω εξομοίωσης. Η ιδέα είναι να γίνει κατανοητός ο τρόπος ελέγχου του σμήνους μέσω των παραμέτρων του συστήματος. Για παράδειγμα, αν ένα συγκεκριμένο σήμα εξασθενεί με κάποιο συγκεκριμένο ρυθμό, ποιος πρέπει να είναι ο ρυθμός αυτός και πόσο επηρεάζει την συλλογική συμπεριφορά του συστήματος;



Εικόνα 1.6 Αποικία μυρμηγκιών

Κεφάλαιο 2ο

2. Ant Colony Optimization (ACO)

2.1 Εισαγωγή

Οι αλγόριθμοι ACO είναι εμπνευσμένοι από τη συμπεριφορά των αποικιών μυρμηγκιών για την αντιμετώπιση διαφόρων προβλημάτων όπως η αναζήτηση τροφής η κατασκευή αποικίας κ.α. Μια από τις βασικές ιδέες πίσω από τη συμπεριφορά αυτή είναι ότι τα έντομα αυτά μπορούν να επικοινωνήσουν μεταξύ τους μέσω της ποσότητας φερομόνης που αφήνουν να διαχέεται στο περιβάλλον τους. Ο τρόπος που επικοινωνούν παρουσιάζει όλα τα χαρακτηριστικά της στιγμεργίας όπως αυτά αναπτύχθηκαν στην παραπάνω ενότητα (§1.2)

Το πρώτο σύστημα που υλοποίησε τον ACO εισήχθη από τον Marco Dorigo το 1992 με το όνομα Ant System (AS). Πρόκειται για έναν **μεθεωρητικό** αλγόριθμο γενικού σκοπού, ο οποίος μπορεί να χρησιμοποιηθεί για την επίλυση NP-complete (όπως το κλασσικό πρόβλημα του Περιπλανώμενου Πωλητή -TSP) και NP-hard προβλημάτων.

2.2 NP-hard προβλήματα

Διαισθητικά η κλάση πολυπλοκότητας NP είναι το σύνολο όλων των προβλημάτων για τα οποία η λύση τους μπορεί να αποδειχτεί ότι είναι σωστή σε πολυωνυμικό χρόνο. Η κλάση πολυπλοκότητας NP είναι το σύνολο όλων των προβλημάτων απόφασης που μπορούν να επιλυθούν σε πολυωνυμικό χρόνο από μια μη-ντετερμινιστική μηχανή Turing. Η κλάση NP εμπεριέχει επίσης πολλά «δύσκολα» προβλήματα για τα οποία δεν υπάρχει γνωστός πολυωνυμικού χρόνου αλγόριθμος, παρά μόνο εκθετικός. Εάν υπάρχει ένας αλγόριθμος πολυωνυμικού χρόνου για κάποιο από αυτά τα προβλήματα, τότε υπάρχει αλγόριθμος πολυωνυμικού χρόνου για όλα τα προβλήματα της κλάσης.

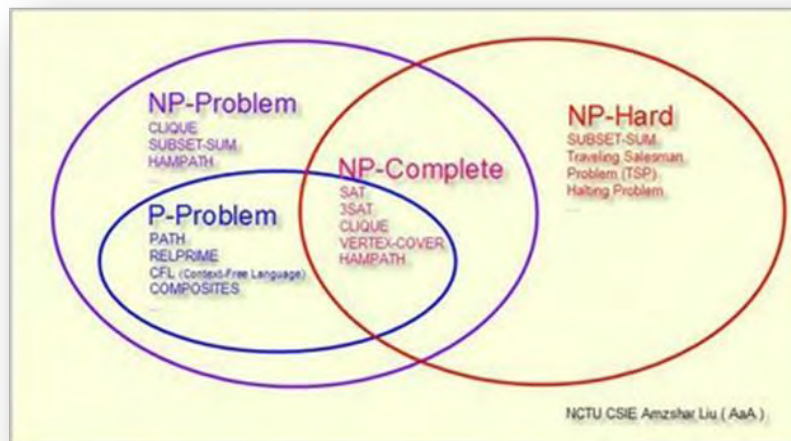
Στην κλάση NP εμπεριέχεται η κλάση πολυπλοκότητας P. Εάν ένα πρόβλημα Π ανήκει στην κλάση P, τότε υπάρχει ένας ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που το επιλύει. Ένα πρόβλημα Π ανήκει στην κλάση NP εάν υπάρχει ένας μη ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου που να το επιλύει. Η κλάση πολυπλοκότητας NP εμπεριέχει την κλάση πολυπλοκότητας P, $P \subseteq NP$. Ένα πολύ σημαντικό ανοικτό πρόβλημα είναι εάν $P = NP$.

Στη συνέχεια θα περιγράψουμε την έννοια της αναγωγής. Ένα πρόβλημα Π_1 ανάγεται πολυωνυμικά σε ένα πρόβλημα Π_2 αν υπάρχει μια συνάρτηση f η οποία απεικονίζει τα στιγμιότυπα του Π_1 στο Π_2 τέτοια ώστε μια λύση για το στιγμιότυπο $f(X)$ του προβλήματος Π_2 αποτελεί λύση για το στιγμιότυπο X του προβλήματος Π_1 ,

για κάθε X . Διαισθητικά αυτό σημαίνει ότι το πρόβλημα Π_1 δεν λύνεται πιο δύσκολα από το Π_2 συν έναν πολυωνυμικό όρο, για το οποίο θα μπορούσαμε να λύσουμε το πρόβλημα Π_1 συνδυάζοντας την συνάρτηση f με τον καλύτερο αλγόριθμο που επιλύει το πρόβλημα Π_2 . Άρα εάν $\Pi_1 \leq \Pi_2$ τότε για την πολυπλοκότητα των προβλημάτων ισχύει: $\text{Complexity}(\Pi_1) \leq \text{Complexity}(\Pi_2) + \text{Polynomial}$. Αφού ορίσαμε την έννοια της αναγωγής μπορούμε να περιγράψουμε τα NP - hard προβλήματα. Ένα πρόβλημα Π είναι NP - hard αν και μόνο αν ισχύει κάποια από τις ισοδύναμες συνθήκες.

1. $\Pi' \leq \Pi$ για κάθε $\Pi' \in \text{NP}$
2. $\Pi \in \text{P} \Rightarrow \text{P} = \text{NP}$
3. Η ύπαρξη ενός ντετερμινιστικού πολυωνυμικού αλγορίθμου για το πρόβλημα Π σημαίνει την ύπαρξη ενός τέτοιου αλγορίθμου για κάθε πρόβλημα που ανήκει στο NP. Ένα πρόβλημα Π είναι N - πλήρες εάν ανήκει στο NP και είναι NP - hard. Για να δείξουμε ότι ένα πρόβλημα είναι NP-πλήρες δείχνουμε ότι το πρόβλημα Π ανήκει στο NP και κάποιο γνωστό NP-πλήρες πρόβλημα Π' μετασχηματίζεται σε αυτό.

Η σημασία της κλάσης NP-Hard έγκειται στο γεγονός ότι αν ανακαλυφθεί ένας αλγόριθμος πολυωνυμικής πολυπλοκότητας $O(n^k)$ που να επιλύει ένα NP-Hard πρόβλημα A τότε όλα τα NP προβλήματα B θα επιλύονται σε πολυωνυμικό χρόνο $O(n^m)$ στο πρόβλημα A και κατόπιν θα λύνουμε το A σε χρόνο $O(n^k)$. Οι πιθανές σχέσεις των συνόλων P, NP, NP-Hard και NP-Complete παρουσιάζονται στο παρακάτω σχήμα (εικόνα 2.1).



Εικόνα 2.1 Κλάσεις NP

Η δυσκολία των NP - Hard προβλημάτων προέρχεται από τα ακόλουθα χαρακτηριστικά:

- ✓ έχουν πολλά διαφορετικά τοπικά βέλτιστα
- ✓ δεν μπορούν να αναλυθούν σε υπό-προβλήματα που να λύνονται ξεχωριστά
- ✓ εμφανίζονται τυχαία στατιστικά (→ οι στατιστικές μέθοδοι δεν λειτουργούν)

2.3 Πρόβλημα του Περιπλανώμενου Πωλητή - TSP

Το πρόβλημα του περιπλανώμενου πωλητή είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης που μελετάται στην θεωρητική επιστήμη των υπολογιστών και στην έρευνα εφαρμογών. Με δεδομένη μία λίστα από πόλεις και των μεταξύ τους αποστάσεων, το ζητούμενο του προβλήματος είναι να βρεθεί η συντομότερη διαδρομή που θα περιλαμβάνει όλες τις πόλεις από μία μόνο φορά.

Το πρόβλημα αρχικά διατυπώθηκε σαν ένα μαθηματικό πρόβλημα το 1930 και είναι ένα από τα πιο εντατικά μελετημένα προβλήματα βελτιστοποίησης με αποτέλεσμα να θεωρείται σημείο αναφοράς. Παρόλη τη μαθηματική δυσκολία του προβλήματος, ένας μεγάλος αριθμός εμπειρικών αλλά και επακριβών μεθόδων είναι γνωστές και έτσι προβλήματα που περιλαμβάνουν δεκάδες χιλιάδες πόλεις μπορούν να λυθούν.

Το πρόβλημα του περιπλανώμενου πωλητή έχει πληθώρα εφαρμογών ακόμη και στην πιο αρχική διατύπωση, όπως στον σχεδιασμό, στην λογιστική, κατασκευή μικροτσίπ. Με παραλλαγές, εμφανίζεται σαν ένα υπο - πρόβλημα πολλών τομέων όπως για παράδειγμα στην γενετική αλληλουχία, σε συστήματα περιορισμένων πόρων κ.α.

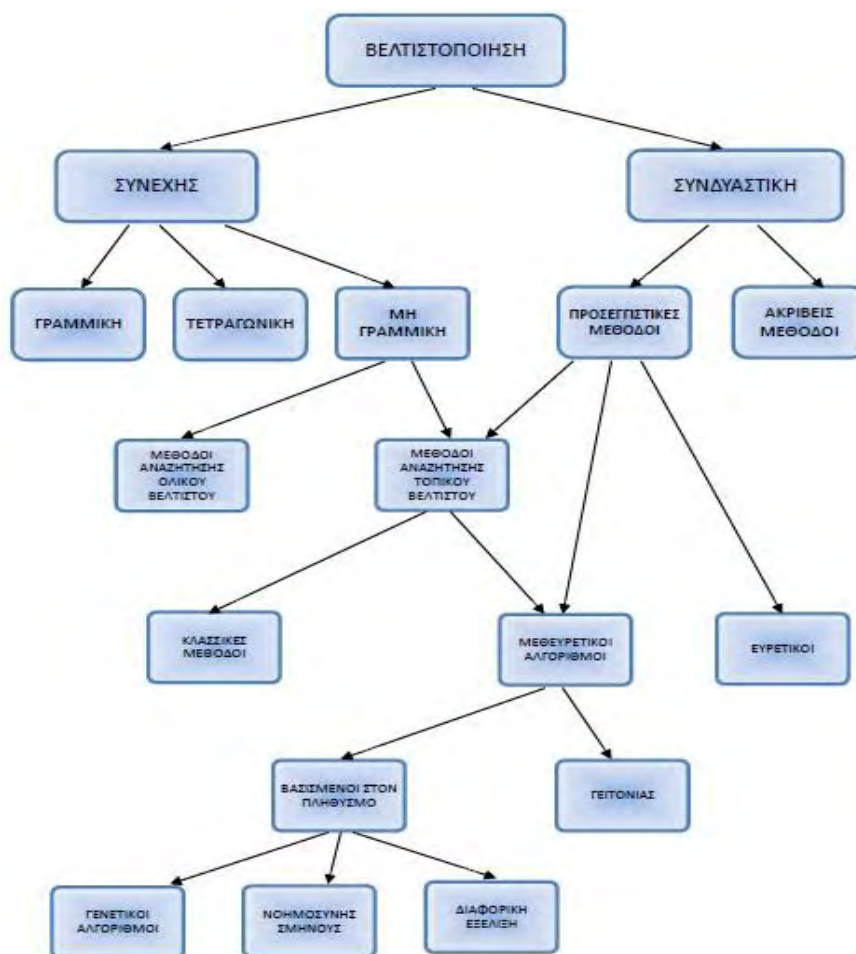
Στην θεωρία της υπολογιστικής πολυπλοκότητας, η εκδοχή του προβλήματος του περιπλανώμενου πωλητή ανήκει στην κλάση των NP-problems, δηλαδή nondeterministic polynomial time problems.

2.4 Μεθευρετικοί (meta-heuristics) αλγόριθμοι

Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Τα τελευταία χρόνια η χρήση τους είναι έντονη για την επίλυση NP – πλήρων προβλημάτων. Χρησιμοποιούνται για να επιλύσουμε προβλήματα για τα οποία γνωρίζουμε πολύ λίγα που μπορούν να μας βοηθήσουν: δε γνωρίζουμε πως θα μοιάζει η βέλτιστη λύση, δε γνωρίζουμε πως θα τη βρούμε με κάποια συγκεκριμένη, δημοφιλή, τακτική και η πλήρη απαρίθμηση των λύσεων είναι αδύνατη, λόγω του εύρους του πεδίου των λύσεων. Όμως, όταν μας δοθεί μια οποιαδήποτε λύση μπορούμε να τη δοκιμάσουμε και να κρίνουμε πόσο ικανοποιητική είναι.

Ένα βασικό χαρακτηριστικό σχεδόν όλων των μεθευρετικών αλγορίθμων είναι η επεξεργασία τυχαίων αναζητήσεων και μεθόδων, και γενικότερα παραδοσιακών αλγορίθμων σαν υποδιαδικασίες τους. Άλλα βασικά χαρακτηριστικά τους, και συγκεκριμένα μιας ολόκληρης οικογένειας μεθευρετικών αλγορίθμων, των γενετικών, είναι ότι πίσω από τη σύλληψή τους κρύβεται ένα φαινόμενο που συναντάμε στη φύση (όπως ο αλγόριθμος των μυρμηγκιών, ACO) μπορούν να μεταφερθούν εύκολα σε παράλληλη μορφή και είναι προσαρμοστικοί αλγόριθμοι. Στοιχεία που συνηθίζονται στην πλειοψηφία τους, είναι: η χρήση ενός πλήθους επαναληπτικών δοκιμών, περιλαμβάνουν ένα ή περισσότερους πράκτορες, λειτουργούν βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού, και περιλαμβάνουν διαδικασίες αυτο-τροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης του προβλήματος.

Ανάλογα με τον αριθμό πρακτόρων που χρησιμοποιούν (έναν ή περισσότερους) μπορούμε να τους διαχωρίσουμε σε δύο κατηγορίες. Μεθευρετικοί βασιζόμενοι στη γειτονιά και μεθευρετικοί βασιζόμενοι στον πληθυσμό. Ένας από αυτούς είναι και ο αλγόριθμος της διαφορικής εξέλιξης ο οποίος ανήκει στη δεύτερη κατηγορία, όπως και η γενετική προσομοίωσή του



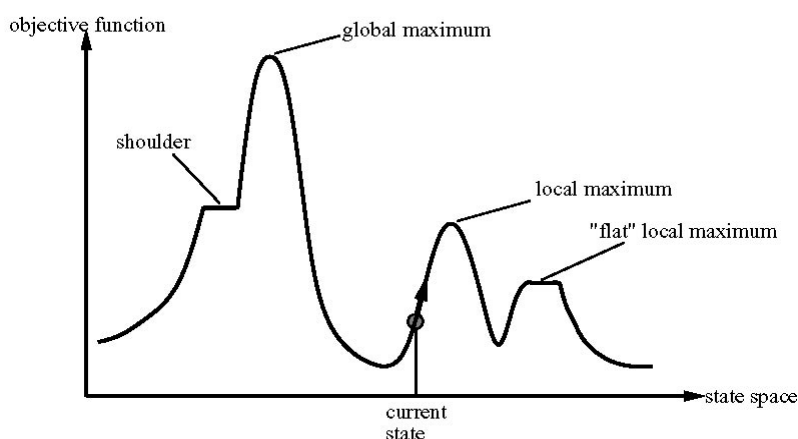
Διάγραμμα 2.1 Κατηγορίες Αλγορίθμων

2.4.1 Αλγόριθμοι τοπικής αναζήτησης

Οι αλγόριθμοι τοπικής αναζήτησης (local search algorithms) όμως ενδιαφέρονται για τις διαδρομές. Λειτουργούν χρησιμοποιώντας μια τρέχουσα κατάσταση και μετακινούνται σε γειτονικές καταστάσεις προσπαθώντας να φτάσουν στην τελική κατάσταση-στόχος. Χρησιμοποιούν λίγη μνήμη (συνήθως σταθερή) και μπορούν να βρουν λύσεις σε πολύ μεγάλους ή και άπειρους (συνεχείς) χώρους.

2.4.1.1 Αναζήτηση με αναρρίχηση λόφων (Hill Climbing)

Ο αλγόριθμος επεκτείνει τα παιδιά της αρχικής (ή τρέχουσας) κατάστασης και αν κάποιο από αυτά έχει μεγαλύτερη αξία τότε μεταβαίνει εκεί και επαναλαμβάνει τη διαδικασία, αλλιώς επιστρέφει την τρέχουσα κατάσταση. Ο αλγόριθμος είναι πολύ οικονομικός σε χρόνο και μνήμη, αλλά παγιδεύεται εύκολα σε τοπικό μέγιστο ή σε οροπέδιο



Γράφημα 2.1 Αναζήτηση με αναρρίχηση λόφων

Για την αποφυγή των τοπικών μέγιστων και των οροπεδίων υπάρχουν πολλές παραλλαγές του αλγορίθμου:

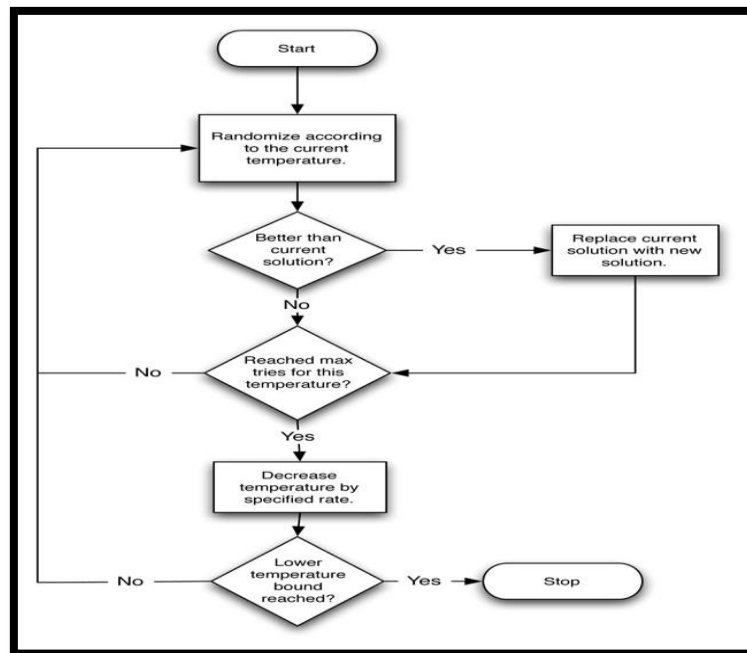
- HC με πλάγια βήματα. Δηλαδή ο πράκτορας κάνει έναν αριθμό πεπερασμένων βημάτων προς κάποια κατεύθυνση ώστε να ξεφύγει από κάποιο εμπόδιο.
- HC με πρώτη επιλογή. Ο αλγόριθμος προχωράει στην πρώτη κατάσταση που είναι καλύτερη από την τρέχουσα, ακόμα κι αν αυτή δεν είναι η καλύτερη από όλα τα παιδιά. Με αυτόν τον τρόπο αποφεύγεται ο "ορειβάτης" να ανεβαίνει με μεγάλη κλίση (που μπορεί να τον παγιδέψει)
- HC με τυχαίες επανεκκινήσεις. Σε πολλούς greedy και τυχαιοποιημένους αλγορίθμους είναι προτιμότερο να γίνονται επανεκκινήσεις. Οι πιθανότητες επιτυχίας της κατάστασης στόχου από p στη μία επανεκκίνηση γίνονται $n \cdot p$, με n επανεκκινήσεις.
- Στοχαστικός HC, ο οποίος διαλέγει την επόμενη κατάσταση στην τύχη, με συντελεστή ανάλογο της αξίας του κάθε παιδιού.

ο αλγόριθμος των βημάτων είναι :

1. Θέσε τρέχουσα κατάσταση την αρχική κατάσταση
2. Αν η τρέχουσα είναι τελική κατάσταση δώσε τη λύση και σταμάτησε
3. Εφάρμοσε τη συνάρτηση διαδόχων για να βρεις διάδοχες καταστάσεις
4. Επέλεξε την καλύτερη διάδοχη κατάσταση σύμφωνα με την ευρετική συνάρτηση
5. Αν η διάδοχη κατάσταση είναι καλύτερη από την τρέχουσα τότε θέσε την ως τρέχουσα και πήγαινε στο 2
6. Αλλιώς σταμάτα στην τρέχουσα κατάσταση (τοπικά καλύτερη)

2.4.1.2 Αναζήτηση με προσομοιωμένη ανόπτηση (Simulated Annealing)

Ο αλγόριθμος προσομοιωμένης ανόπτησης¹ χρησιμοποιεί "κατηφορικές" κινήσεις και επιλέγει μετάβαση μεταξύ τυχαίων γειτόνων. Αν η κίνηση βελτιώνει την κατάσταση τότε γίνεται αποδεκτή, αλλιώς γίνεται αποδεκτή με πιθανότητα $\exp(\Delta E/T)$. Όπου ΔE είναι η διαφορά της τρέχουσας κατάστασης από την νέα και T είναι ένας μετρητής που μειώνεται με κάποιο ρυθμό. Έτσι "κακές" ή κατηφορικές κινήσεις ενδέχεται να γίνουν αποδεκτές στην αρχή, αλλά όχι προς το τέλος. Αν το T μειώνεται με αργό ρυθμό ο αλγόριθμος φτάνει στη βέλτιστη λύση με πιθανότητα σχεδόν 1.



Διάγραμμα 2.1 Γράφος αποφάσεων Simulated Annealing

¹ Ανόπτηση στη μεταλλουργεία είναι η διαδικασία στην οποία θερμαίνεται ένα μέταλλο σε υψηλή θερμοκρασία ώστε να του δοθεί μορφή και μετά ψύχεται βαθμιαία ώστε το υλικό να καταλήξει σε κρυσταλλική κατάσταση χαμηλής ενέργειας (δλδ. γίνεται πολύ πιο άκαμπτο). Το ρόλο της θερμοκρασίας που πέφτει σταθερά κάνει το T στον αλγόριθμο

2.4.1.3 Τοπική ακτινική αναζήτηση (Beam search)

Ο αλγόριθμος τοπικής ακτινικής αναζήτησης ξεκινά με k τυχαιοποιημένες καταστάσεις και παράγει σε κάθε επανάληψη τους διαδόχους τους. Στη συνέχεια κρατάει τις k καλύτερες καταστάσεις και επαναλαμβάνει. Αν και φαίνεται για έναν αλγόριθμο που απλά κάνει k επανεκκινήσεις, στην πραγματικότητα μεταβιβάζει χρήσιμες πληροφορίες μεταξύ των k παράλληλων νημάτων αναζήτησης. Μπορεί να μετατραπεί και αυτός ο αλγόριθμος σε στοχαστικό, διαλέγοντας τους k διαδόχους με κάποια τυχειότητα.

2.4.1.4 Αναζήτηση με γενετικό αλγόριθμο (Genetic Algorithm)

Ο γενετικός αλγόριθμος είναι μια παραλλαγή του στοχαστικού αλγόριθμου ακτινικής αναζήτησης, με τη διαφορά της συμμετοχής δύο γονικών καταστάσεων και όχι μιας μεμονωμένης κατάστασης. Η διαδικασία της λειτουργίας του αλγόριθμου είναι εμπνευσμένη από την γενετική. Στην αρχή διαλέγονται k καταστάσεις- "χρωμοσώματα" που ονομάζονται "πληθυσμός". Κάθε κατάσταση συμβολίζεται με μία γραμματοσειρά, ανάλογα με τις ανάγκες του προβλήματος. Με τη χρήση μιας συνάρτησης καταλληλότητας (fitness function), επιλέγονται τα χρωμοσώματα και διασταυρώνονται. Μιμούμενοι τη φυσική διασταύρωση των γονιδίων, στα νέα χρωμοσώματα γίνονται μεταλλάξεις, οι οποίες φροντίζουν για την ποικιλότητα του νέου πληθυσμού.

2.4.1.5 Αναζήτηση απαγορευμένων καταστάσεων (tabu search, TS)

Πρόκειται για μια μέθοδο τοπικής αναζήτησης, η οποία προσπαθεί να ξεπεράσει τον εγκλωβισμό σε τοπικά ελάχιστα της αντικειμενικής συνάρτησης. Η μέθοδος αυτή συνήθως δεν χρησιμοποιείται μεμονωμένα αλλά σε συνδυασμό με κάποιον άλλο στοχαστικό αλγόριθμο, τον οποίον και κατευθύνει. Για αυτό τον λόγο άλλωστε καλείται και μεθευρετικός (metaheuristic) αλγόριθμος.

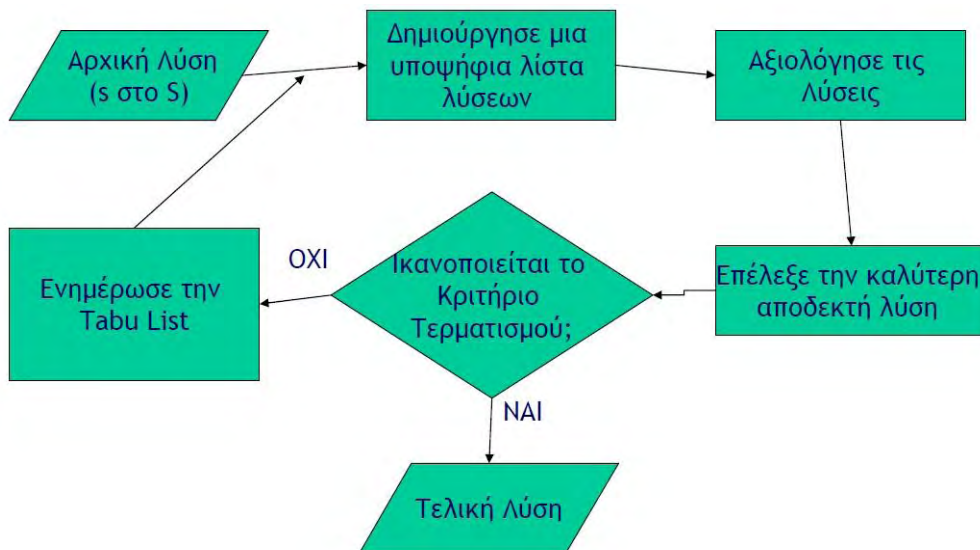
Ένας τρόπος για την επίτευξη του στόχου αυτού είναι η αλλαγή της τεχνικής ενημέρωσης της λύσης. Σε αντίθεση με άλλες μεθόδους βελτίωσης, σε κάθε βήμα της επαναληπτικής διαδικασίας της επιτρέπεται η επιλογή μιας λύσης όχι μόνο με μικρότερη, αλλά και με μεγαλύτερη τιμή από αυτή της υπάρχουσας λύσης. Στην αναζήτηση απαγορευμένων καταστάσεων, βασική ιδέα είναι ο περιορισμός των κατευθύνσεων αναζήτησης σε κάθε βήμα της επανάληψης, ώστε να επιτευχθεί η εύρεση καλής λύσης με αποδοτικό τρόπο. Η μέθοδος "θυμάται" ποιες λύσεις έχουν ήδη αναζητηθεί, ώστε να αναζητήσει τις περισσότερο υποσχόμενες κατευθύνσεις.

Τα βασικά χαρακτηριστικά γνωρίσματα μιας αναζήτησης απαγορευμένων καταστάσεων είναι η δομή της γειτονιάς (neighborhood structure), οι κινήσεις (moves), η λίστα απαγορευμένων καταστάσεων (tabu list) και το κριτήριο φιλοδοξίας (aspiration criterion). Μια κίνηση είναι η διαδικασία εύρεσης μιας γειτονικής λύσης δεδομένης μιας υπάρχουσας λύσης. Η λίστα απαγορευμένων

καταστάσεων περιλαμβάνει ένα σύνολο κινήσεων, οι οποίες δεν επιτρέπεται να συμπεριληφθούν στη λύση που αναζητείται. Η λίστα αυτή ενημερώνεται διαρκώς με νέες εισόδους ή διαγραφές σε αυτήν. Το κριτήριο φιλοδοξίας είναι μια προϋπόθεση, την οποία αν ικανοποιεί μια κίνηση τότε η τελευταία διαγράφεται από τη λίστα απαγορευμένων καταστάσεων .

ο αλγόριθμος των βημάτων είναι :

1. Κατασκευάσε μια αρχική λύση s στο χώρο των λύσεων S . Θέσε $s^* = s$ και $t=0$, όπου ως s^* συμβολίζεται η συνολικά καλύτερη λύση (που έχει βρεθεί μετά το σύνολο όλων των επαναλήψεων που έχουν διεξαχθεί) και ως t συμβολίζεται η επανάληψη του αλγορίθμου tabu search
2. Θέσε $t = t+1$ και παρήγαγε ένα υποσύνολο V^* γειτονικών λύσεων στη γειτονία $N(s)$, στην επανάληψη t , οι οποίες είτε δεν παράγονται από tabu χαρακτηριστικά είτε μία από αυτές ικανοποιεί το κριτήριο υπέρβασης
3. Επέλεξε την καλύτερη λύση s^+ του V^* και θέσε $s = s^+$
4. Εάν η αντικειμενική συνάρτηση $f(s) < f(s^*)$ τότε θέσε $s^* = s$
5. Ενημέρωσε την Tabu List
6. Εάν ενεργοποιείται το κριτήριο τερματισμού, τότε σταμάτησε. Αλλιώς πήγαινε στο Βήμα 2



Σχήμα 2.1 Διάγραμμα ροής Tabu Search

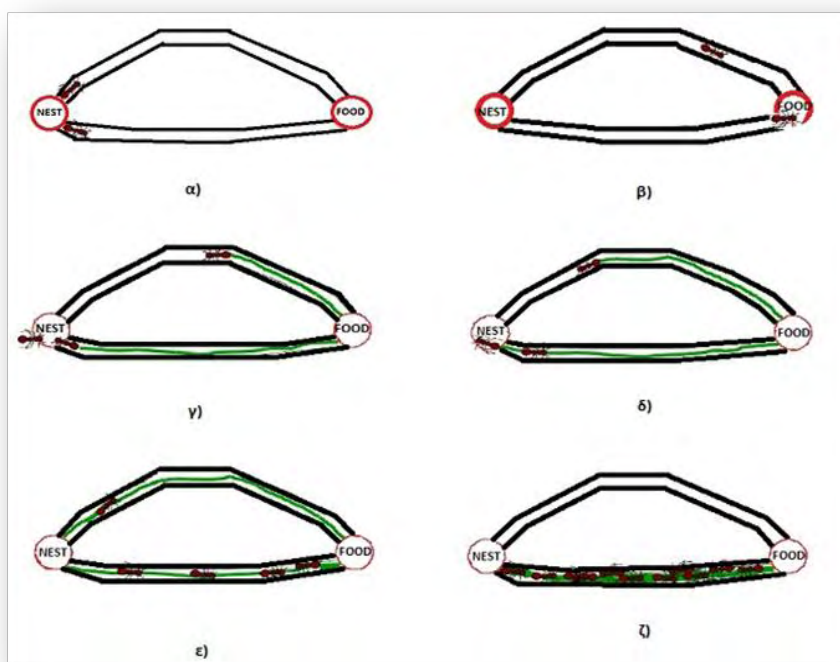
2.5 Πραγματικά μυρμήγκια

Το πεδίο των αλγορίθμων τύπου αποικίας μυρμηγκιών μελετά μοντέλα τα οποία προέρχονται από την παρατήρηση της συμπεριφοράς πραγματικών μυρμηγκιών, και χρησιμοποιεί αυτά τα μοντέλα ως πηγή έμπνευσης για τον σχεδιασμό πρωτότυπων αλγορίθμων στην επίλυση προβλημάτων βελτιστοποίησης και καταναεμημένου ελέγχου. Η κεντρική ιδέα είναι ότι οι αρχές αυτό-οργάνωσης που επιτρέπουν μία τόσο υψηλά συντονισμένη συμπεριφορά στα πραγματικά μυρμήγκια μπορεί να αξιοποιηθεί στο συντονισμό πληθυσμών ψηφιακών πρακτόρων οι οποίοι συνεργάζονται για την επίλυση υπολογιστικών προβλημάτων.

2.5.1 Αναζήτηση τροφής

Τα μυρμήγκια είναι κοινωνικά έντομα και η συμπεριφορά τους εστιάζεται στην επιβίωση της αποικίας και όχι στην επιβίωση του ατόμου. Περαιτέρω, αρκετά νωρίς στην έρευνα είχε παρατηρηθεί ότι το μεγαλύτερο ποσοστό της επικοινωνίας των ατόμων με το περιβάλλον βασίζεται στη χρήση χημικών που παράγονται από τα μυρμήγκια. Αυτά τα χημικά ονομάζονται φερομόνες. Καθώς κινούνται προς τις πηγές τροφής και αντιστρόφως, τα μυρμήγκια εναποθέτουν φερομόνες στο έδαφος, σχηματίζοντας με αυτό το τρόπο μονοπάτια από φερομόνη. Τα μυρμήγκια μπορούν να μυρίσουν τη φερομόνη και τείνουν να επιλέγουν, πιθανοκρατικά, μονοπάτια με υψηλότερη συγκέντρωση φερομόνης. Έχει αποδειχθεί πειραματικά ότι σε κάποια μυρμήγκια η συμπεριφορά σχηματισμού μονοπατιών φερομόνης και η ακολούθηση αυτών μπορεί να οδηγήσει στον εντοπισμό συντομότερων μονοπατιών.

Για παράδειγμα δύο μυρμήγκια ξεκινούν ταυτόχρονα (α) την αναζήτησή τους για τροφή. Το πρώτο ακολουθεί το πάνω μονοπάτι και το δεύτερο το κάτω μονοπάτι. Το δεύτερο φτάνει νωρίτερα στην τροφή (β) και αφού πάρει ένα μέρος της παίρνει τον δρόμο του γυρισμού για την φωλιά αφήνοντας πίσω ίχνη με την αναδυόμενη φερομόνη του για να τα ακολουθήσει το επόμενο μυρμήγκι. Και το άλλο μυρμήγκι ανακαλύπτει την τροφή και παίρνει με την σειρά του τον δρόμο του γυρισμού αφήνοντας και αυτό την δική του φερομόνη, ενώ ένα τρίτο ξεκινάει την δική του αναζήτηση (γ). Έχοντας να διαλέξει μεταξύ δύο μονοπατιών επιλέγει αυτό το οποίο έχει ίχνη φερομόνης που άφησε το προηγούμενο, μιας και ακολουθώντας το, είναι σίγουρο πως θα βρει τροφή (δ).



Εικόνα 1.2 Αναζήτηση τροφής

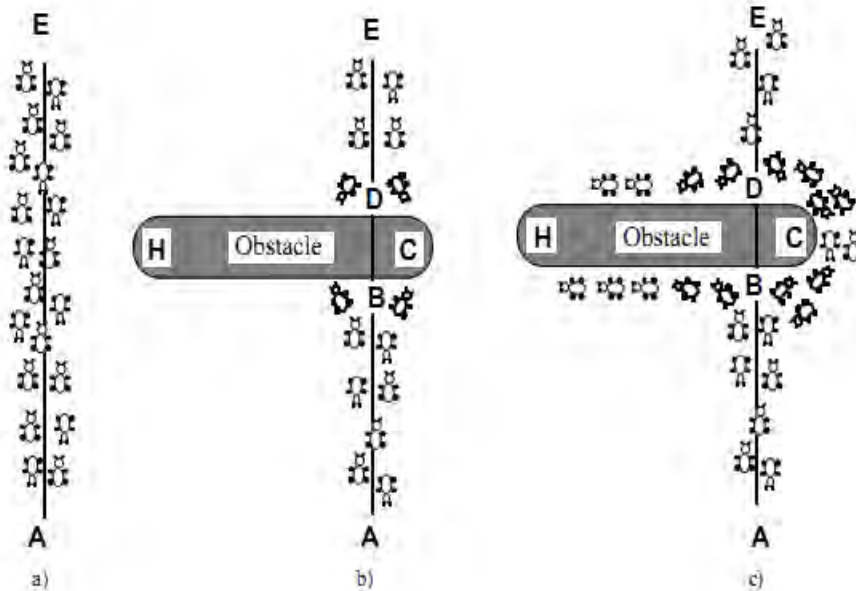
Ανακαλύπτοντας την τροφή παίρνει ένα κομμάτι από αυτή και γυρίζει στην φωλιά αφήνοντας και αυτό φερομόνη που ενισχύει την ήδη υπάρχουσα. Όλο και περισσότερα μυρμήγκια επιλέγουν το συγκεκριμένο μονοπάτι και αυτό γιατί το δεύτερο μονοπάτι είναι συντομότερο μιας και τα μυρμήγκια επιστρέφουν συντομότερα στην φωλιά ενισχύοντας με περισσότερη φερομόνη το δεύτερο μονοπάτι. (ε). Έτσι κάποια στιγμή όλα τα μυρμήγκια θα επιλέγουν ένα μονοπάτι ενώ η φερομόνη από το δεύτερο θα εξασθενίσει με την πάροδο του χρόνου, βγάζοντας το συγκεκριμένο μονοπάτι από τον “χάρτη” αναζήτησης τροφής των μυρμηγκιών (ζ).

2.5.2 Αποφυγή Εμποδίων

Άξιο έρευνας αποτελεί και ο τρόπος με τον οποίο τα μυρμήγκια αποφεύγουν τα εμπόδια που εμφανίζονται στον δρόμο τους. Στην εικόνα παρουσιάζεται η κίνηση αυτή και αναλύεται ο τρόπος αντίδρασης που πρέπει να υιοθετήσουν τα μυρμήγκια καθώς και οι αποφάσεις που πρέπει να λάβουν.

Αρχικά όπως φαίνεται στο σχήμα **a** της εικόνας 2.3 υπάρχει ένα μονοπάτι το οποίο ακολουθούν τα μυρμήγκια από τη φωλιά A στην πηγή της τροφής E και το αντίστροφο. Κάποια στιγμή εμφανίζεται ένα εμπόδιο που δεν επιτρέπει στα μυρμήγκια να ακολουθήσουν το μονοπάτι που ακολουθούσαν ως τώρα, όπως φαίνεται στο σχήμα **b** της εικόνας 2.3.

Κατά συνέπεια τα μυρμήγκια που κατευθύνονται στο σημείο E πρέπει να αποφασίσουν στο σημείο B αν θα κινηθούν προς τα δεξιά ή τα αριστερά. Αντίστοιχη απόφαση πρέπει να πάρουν και τα μυρμήγκια που κινούνται προς την αντίθετη κατεύθυνση, στο σημείο D. Η απόφαση για το ποιο μονοπάτι θα ακολουθήσουν επηρεάζεται από την ποσότητα φερομόνης που έχει εναποτεθεί από τα μυρμήγκια που έχουν ακολουθήσει προηγουμένως τα διάφορα μονοπάτια. Υψηλότερα επίπεδα φερομόνης σε κάποιο μονοπάτι αυξάνουν την πιθανότητα να επιλεγεί εκείνο έναντι του άλλου διαθέσιμου μονοπατιού. Το πρώτο μυρμήγκι που θα φτάσει στο σημείο B (ή D) έχει ίσες πιθανότητες να επιλέξει είτε το αριστερό είτε το δεξί μονοπάτι (καθώς δεν υπάρχει ίχνος φερομόνης σε κανένα από τα 2 μονοπάτια). Εξαιτίας του γεγονότος ότι το μονοπάτι που περνάει από το σημείο C είναι μικρότερο σε μήκος από το μονοπάτι που διέρχεται από το σημείο H, το πρώτο μυρμήγκι θα ακολουθήσει το μονοπάτι BCD θα φτάσει στο σημείο D πριν από το αντίστοιχο πρώτο μυρμήγκι που ακολουθήσει το μονοπάτι BHD.



Εικόνα 2.3 Αποφυγή εμποδίων

Συνεπώς ένα μυρμήγκι που κινείται από το σημείο E στο σημείο D θα εντοπίσει ισχυρότερο ίχνος φερομόνης στο μονοπάτι DCB, το οποίο θα έχουν αφήσει τα μισά από το σύνολο των μυρμηγκιών, ο αριθμός δηλαδή των μυρμηγκιών που προτίμησαν τυχαία το μονοπάτι DCBA καθώς και όσα μυρμήγκια επέστρεψαν από το μονοπάτι BCD. Αυτό έχει ως αποτέλεσμα να είναι πιθανότερο το μυρμήγκι να προτιμήσει το μονοπάτι DCB από το DHB όπως φαίνεται και στο σχήμα c της εικόνας 2.3. Κατά συνέπεια ο αριθμός των μυρμηγκιών που ακολουθούν το μονοπάτι BCD στη μονάδα του χρόνου είναι μεγαλύτερος από τον αριθμό των μυρμηγκιών που ακολουθούν το μονοπάτι BHD στον ίδιο χρόνο. Αυτό προκαλεί ταχύτερη αύξηση της ποσότητας της φερομόνης στο συντομότερο μονοπάτι, οπότε η πιθανότητα με την οποία επιλέγεται το συντομότερο μονοπάτι αυξάνεται και αμέσως μετά το μεγαλύτερο κομμάτι του συνόλου των μυρμηγκιών ακολουθεί το μονοπάτι

2.6 Ψηφιακά μυρμήγκια

Προκειμένου να μετατρέψουμε τα φυσικά βιολογικά μοντέλα σε τεχνητά “αλγοριθμικά”, δηλαδή σε υπολογιστικά εργαλεία για την επίλυση προβλημάτων βελτιστοποίησης, θα πρέπει να κάνουμε ορισμένες παραδοχές. Προφανώς, τα “ψηφιακά” μυρμήγκια θα αποτελούν μια απλοποίηση των πραγματικών μυρμηγκιών αλλά και θα εμπλουτιστούν με επιπρόσθετες ιδιότητες που δεν εμφανίζονται στα φυσικά μυρμήγκια και θα καθορίζονται από το αντίστοιχο πρόβλημα προς λύση ή από τις ανάγκες του χρήστη για γρήγορη σύγκλιση κτλ.

Οι ομοιότητες των “ψηφιακών” μυρμηγκιών με τα φυσικά έχουν ως εξής:

- **Η αποικία.** Όπως στα φυσικά μυρμήγκια έτσι και στα “ψηφιακά” υπάρχει μια αποικία ατόμων τα οποία συνεργάζονται προκειμένου να δημιουργήσουν μια συγκεκριμένη δομή στο σύστημά τους. Η δομή αυτή στα υπό μελέτη προβλήματα στην παρούσα εργασία αντιστοιχεί στην εύρεση βέλτιστης λύσης. Ο πληθυσμός των «ψηφιακών» μυρμηγκιών είναι μια παράμετρος η οποία είναι σε άμεση συνάρτηση με το είδος του προβλήματος.
- **Η φερομόνη.** Ορισμένες “ποικιλίες” πραγματικών μυρμηγκιών εναποθέτουν φερομόνη στο έδαφος κατά την κίνησή τους. Ομοίως και τα “ψηφιακά” μυρμήγκια θα εναποθέτουν στις διάφορες θέσεις τις οποίες επισκέπτονται μια συγκεκριμένη αριθμητική πληροφορία, την “ψηφιακή” φερομόνη, οπότε και θα δημιουργούνται μονοπάτια φερομόνης (pheromone trails). Αυτή η αριθμητική πληροφορία θα εξαρτάται από την επίδοση και την προηγούμενη ιστορία του «ψηφιακού» μυρμηγκιού και θα είναι προσβάσιμη και από τα υπόλοιπα μέλη της αποικίας εγκαθιδρύοντας με αυτόν τον τρόπο στιγμεργετική (stigmergetic) επικοινωνία. Τέλος, η “ψηφιακή” φερομόνη όπως και η πραγματική θα πρέπει να υφίσταται εξάτμιση έτσι ώστε να υπάρχει αρνητική ανάδραση και να μην εγκλωβίζονται τα άτομα σε τοπικά βέλτιστα.
- **Η απόφαση.** Όπως τα φυσικά έτσι και τα “ψηφιακά” μυρμήγκια αποφασίζουν την επόμενη κίνησή τους με κάποια στοχαστικότητα. Τα δεδομένα που έχουν στη διάθεσή τους είναι καθαρά τοπικά στο χώρο και το χρόνο. Αναλυτικότερα, τα μυρμήγκια, είτε πραγματικά είτε “ψηφιακά”, μυρμήγκια δεν έχουν γνώση του τι συμβαίνει σε άλλο σημείο όπου δραστηριοποιούνται μυρμήγκια της αποικίας αλλά ούτε και τη γενικότερη κατάσταση της αποικίας. Επίσης, δεν γνωρίζουν τι έγινε τις προηγούμενες χρονικές στιγμές στο σημείο όπου βρίσκονται αλλά ούτε μπορούν να προβλέψουν μελλοντικές καταστάσεις.

Οι επιπλέον ικανότητες που προαιρετικά μπορούν να προστεθούν στα “ψηφιακά” μυρμήγκια έχουν ως εξής:

1. Ύπαρξη “μνήμης”: για να θυμούνται τις προηγούμενες πράξεις τους (πχ Tabu list στο TSP) ή άλλες χρήσιμες πληροφορίες.
2. Ύπαρξη “όρασης” ή άλλων “αισθητηρίων”: ώστε να εκμεταλλεύονται καλύτερα τις πληροφορίες του περιβάλλοντος (πχ. να μπορούν να υπολογίσουν εκ των προτέρων τις αποστάσεις των πόλεων στο TSP και όχι αφού έχουν ολοκληρώσει μια διαδρομή).
3. Διακριτοποίηση του χώρου και του χρόνου: για την επίλυση διακριτών προβλημάτων. Για παράδειγμα, οι κινήσεις των μυρμηγκιών στο TSP είναι διακριτές από πόλη σε πόλη και πραγματοποιούνται σε διακριτές χρονικές στιγμές.
4. Εναπόθεση φερομόνης σε διαφορετικές χρονικές στιγμές: ενώ τα φυσικά μυρμήγκια εναποθέτουν συνεχώς φερομόνη, τα “ψηφιακά” το κάνουν συνήθως αφού ολοκληρώσουν τις κινήσεις τους.
5. Χρησιμοποίηση αλγορίθμων για τη βελτίωση της απόδοσης του συστήματος: μπορεί να χρησιμοποιηθούν αλγόριθμοι τοπικής βελτιστοποίησης κτλ.

2.7 Αλγόριθμοι ACO

Οι αλγόριθμοι που προέρχονται από τη μελέτη της συμπεριφοράς των αποικιών μυρμηγκιών, έχουν ως στόχο την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης και όχι την εξομοίωση της λειτουργίας των αποικιών των μυρμηγκιών και ως εκ τούτου παρουσιάζουν ορισμένες σημαντικές διαφορές από τα χαρακτηριστικά που παρουσιάζουν τα ανάλογα συστήματα στη φύση: τα τεχνητά μυρμήγκια έχουν κάποια διαθέσιμη μνήμη τα και χρησιμοποιούν σε κάποιο βαθμό την όρασή τους .

Η κάθε λύση για το TSP με χρήση αυτού του αλγορίθμου σχηματίζεται με τη διαδοχική μετάβαση των τεχνητών μυρμηγκιών-πρακτόρων από τη μια πόλη στην άλλη με κάποια πιθανότητα . Το «ταξίδι» ολοκληρώνεται όταν το τεχνητό μυρμήγκι επιστρέψει στην αρχική πόλη. Τότε βαθμολογείται η λύση που έχει επιτύχει με κριτήριο το συνολικό μήκος της διαδρομής και προστίθεται η ανάλογη φερομόνη στο μονοπάτι που ακολούθησε. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε άτομο της αποικίας μέχρι να συμπληρωθεί ο ζητούμενος αριθμός επαναλήψεων.

Αρχικά τα μυρμήγκια τοποθετούνται είτε τυχαία στις πόλεις είτε το καθένα σε μια διαφορετική πόλη ως αφετηρία της διαδρομής τους. Η πόλη αυτή καταγράφεται ως η αφετηρία στη λίστα της μνήμης κάθε μυρμηγκιού έτσι ώστε να αποκλειστεί από τις επόμενες επιλογές του. Η επιλογή της επόμενης πόλης την οποία θα επισκεφθεί κάθε μυρμήγκι ορίζεται από μια πιθανότητα μετάβασης η οποία εξαρτάται από την «ορατότητα» του κάθε μυρμηγκιού (ορατότητα ορίζεται ως το

αντίστροφο της απόστασης μεταξύ δύο πόλεων και άρα εισάγει μεγάλη πιθανότητα στον κανόνα μετάβασης όσον αφορά τις κοντινές πόλεις) και στην ποσότητα της φερομόνης που έχει εναποτεθεί στην κάθε διαδρομή εκφράζοντας την εμπειρία της υπόλοιπης αποικίας.

Αφού όλα τα μυρμήγκια ολοκληρώσουν τα “ταξίδια” τους σε κάθε επανάληψη, τότε προστίθεται φερομόνη σε κάθε μονοπάτι ανάλογα με την επίδοση του κάθε μυρμηγκιού. Σημαντικό στοιχείο στην όλη διαδικασία είναι και η εισαγωγή ενός παράγοντα σχετικού με την εξάτμιση της φερομόνης. Κατά αυτόν τον τρόπο αποφεύγεται η ενίσχυση των αρχικών τυχαίων διακυμάνσεων.

Τα βήματα του αλγορίθμου ACO είναι τα ακόλουθα:

- ✓ Βήμα 1ο: Θέτουμε τη φερομόνη στην αρχική της τιμή (είτε σε μια πολύ μικρή τιμή είτε σε μια τυχαία τιμή) σε όλα τα μονοπάτια που συνδέουν τις πόλεις μεταξύ τους. Υπολογίζουμε όλες τις αποστάσεις των πόλεων. Επιλέγουμε (τυχαία) την πόλη-αφετηρία της διαδρομής κάθε μυρμηγκιού και την τοποθετούμε στην λίστα της μνήμης του.
- ✓ Βήμα 2ο: Για κάθε μυρμήγκι, επιλέγουμε την επόμενη πόλη που θα επισκεφτεί με βάση τον τυχαίο αναλογικό κανόνα μετάβασης μέχρι να ολοκληρώσει τη διαδρομή του αφού έχει περάσει από κάθε πόλη μια μόνο φορά και τελικά να επιστρέψει στην αφετηρία του.
- ✓ Βήμα 3ο: Καταγράφουμε την καλύτερη διαδρομή που βρέθηκε.
- ✓ Βήμα 4ο: Ανανεώνουμε τη φερομόνη στα μονοπάτια που επισκέφτηκαν τα μυρμήγκια με βάση τον κανόνα ανανέωσης της φερομόνης.
- ✓ Βήμα 5ο: Επαναλαμβάνουμε τη διαδικασία από το βήμα 2 έως ότου ολοκληρωθεί ένας συγκεκριμένος αριθμός επαναλήψεων ή επιτευχθεί ένα κριτήριο σύγκλισης.

2.7.1 Δομή αλγορίθμου

Η ACO είναι μια στοχαστική αναζήτηση λύσης σε ένα συνδυαστικό πρόβλημα με πυρήνα το μοντέλο απόδοσης φερομόνης. Το πρώτο βήμα όμως της εφαρμογής του αλγορίθμου είναι η μοντελοποίηση του προβλήματος. Το μοντέλο $P(S, \Omega, f)$ ενός συνδυαστικού προβλήματος αποτελείται από :

- Ένα χώρο αναζήτησης S ορισμένο από ένα πεπερασμένο αριθμό διακριτών μεταβλητών απόφασης
- Ένα σύνολο περιορισμών Ω
- Μία συνάρτηση $f : S \rightarrow R_0^+$ της οποίας στόχος είναι να βρεθεί το ελάχιστο.

Ο χώρος αναζήτησης ορίζεται ως εξής:

Με δεδομένο ένα σύνολο διακριτών μεταβλητών X_i , $i = 1, \dots, n$ με τιμές v_i^j που ανήκουν σε ένα πίνακα τιμών της μορφής $D^i = \{v_i^1, \dots, v_i^{D_i}\}$, η απόδοση των τιμών v_i^j στο X_i συμβολίζεται με $X_i \leftarrow v_i^j$. Μια λύση s είναι εφικτή όταν όλες οι αποδόσεις τιμών στις μεταβλητές ικανοποιούν τους περιορισμούς Ω . Μια λύση s^* λέγεται ολικό ελάχιστο όταν για κάθε s που ανήκει στο S ισχύει $f(s^*) \leq f(s)$. Το σύνολο των ολικών ελάχιστων συμβολίζεται με S^* . Στόχος του ACO είναι η εύρεση ενός από τα ολικά ελάχιστα. Η απόδοση τιμής σε μια μεταβλητή συμβολίζεται με το c_{ij} το οποίο ονομάζεται τμήμα λύσης. Στη συνέχεια στο κάθε τμήμα λύσης (c_{ij}) αποδίδεται μια τιμή επιπέδου φερομόνης (συμβολισμένη με τ_{ij}). Η τιμή αυτή ανανεώνεται από τον ACO και καθορίζει την πιθανότητα επιλογής ενός τμήματος λύσης.

Στον ACO τα τεχνητά μυρμήγκια κατασκευάζουν μια λύση διασχίζοντας τον γράφο του συνδυαστικού προβλήματος, $G_c(V,E)$, ο οποίος αποτελείται από μια συλλογή κορυφών V και μια συλλογή ακμών E . Τα τμήματα λύσης c αντιστοιχίζονται είτε στις κορυφές είτε στις ακμές του γράφου. Τα τεχνητά μυρμήγκια διασχίζουν τον γράφο από κορυφή σε κορυφή μέσω των ακμών δημιουργώντας σταδιακά μια λύση επιλέγοντας τα τμήματα λύσης c που αντιστοιχούν στις ακμές και κορυφές που διέσχισαν. Επιπλέον, εναποθέτουν σε κάθε c μια ποσότητα φερομόνης η οποία μπορεί να εξαρτάται από διάφορους παράγοντες ανάλογα με την εκάστοτε υλοποίηση του αλγορίθμου. Επόμενα μυρμήγκια χρησιμοποιώντας τα επίπεδα φερομόνης οδηγούνται στον γράφο προς πιο ελπιδοφόρες περιοχές λύσης.

Μια βασική μορφή του αλγόριθμου σε ψευδογλώσσα είναι η εξής :

Algorithm 1 The Ant Colony Optimization Metaheuristic

1. **input** : ένα μοντέλο $P(S,\Omega,f)$ ενός συνδυαστικού προβλήματος
 2. InitializePheromoneValues(T)
 3. **while** δεν έχουν ικανοποιηθεί οι συνθήκες τερματισμού
 4. ConstructAntSolutions
 5. ApplyLocalSearch {προαιρετικό}
 6. UpdatePheromones
 7. **end_while**
-

➤ **InitializePheromoneValues(T)**

Η συνάρτηση που αρχικοποιεί τα επίπεδα φερομόνης σε όλο το γράφο με μια αρχική σταθερά $c > 0$.

➤ **ConstructAntSolutions**

Ένα σύνολο από m τεχνητά μυρμήγκια κατασκευάζει μια λύση επιλέγοντας στοιχεία από το C (το πεπερασμένο σύνολο των τμημάτων λύσης). Μια κατασκευή λύσης ξεκινάει από μια κενή λύση $s^p = 0$ και προσθέτει σε κάθε βήμα ένα τμήμα λύσεων c_{ij} . Αυτό το τμήμα που επεκτείνει τη λύση πρέπει να είναι εφικτό, δηλαδή ανήκει στο σύνολο των τμημάτων λύσης που δεν παραβιάζει κανέναν κανόνα του Ω .

Η επιλογή της επόμενης λύσης έχει ένα στοχαστικό παράγοντα, δηλαδή εξαρτάται πιθανοτικά από το επίπεδο φερομόνης τ_{ij} που αντιστοιχεί σ' αυτό. Ο ακριβής κανόνας είναι χαρακτηριστικό της κάθε διαφοροποίησης του ACO αλλά αντλεί έμπνευση από τον κανόνα των πραγματικών μυρμηγκιών που εξηγήθηκε παραπάνω.

➤ **ApplyLocalSearch**

Αποτελεί ένα προαιρετικό βήμα για την εκτέλεση του αλγορίθμου, αλλά συναντάται σχεδόν σε κάθε πραγματική υλοποίησή του καθώς ανεβάζει την απόδοσή του. Συχνά ονομάζονται *daemon actions*, αλλάζουν ανάλογα με τις ιδιαιτερότητες του προβλήματος και κάνουν συγκεντρωτικές ενέργειες. Η πιο συχνή χρήση των *daemon actions* είναι η τοπική αναζήτηση, η οποία αναζητά τον χώρο των λύσεων στη γειτονιά της κατασκευασμένης λύσης βελτιώνοντας την και χρησιμοποιώντας αυτήν την βελτιωμένη έκδοση για την ανανέωση των επιπέδων φερομόνης.

➤ **UpdatePheromones**

Ο ρόλος αυτής της συνάρτησης είναι να αυξήσει τα επίπεδα φερομόνης των υποσχόμενων λύσεων, δηλαδή αυτών με καλή απόδοση, ώστε να ελκύσει μελλοντικά μυρμήγκια προς την περιοχή τους, αλλά και να μειώσει τα επίπεδα φερομόνης στις λύσεις με μικρότερη απόδοση. Αυτό το καταφέρνει με δύο μηχανισμούς :

- Εξάτμιση φερομόνης: Μείωση όλων των τιμών της φερομόνης σε κάθε επανάληψη.
- Αύξηση των τιμών φερομόνης στο επιλεγμένο σύνολο “καλών” λύσεων S_{upd} .

Έτσι σχηματίζεται η εξής εξίσωση:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{s \in S_{\text{upd}}} F(s)$$

Όπου :

- S_{upd} το σύνολο λύσεων που επιλέγεται για την ανανέωση της φερομόνης τους
- $\rho \in (0,1]$ παράμετρος που λέγεται συντελεστής εξάτμισης
- $F : S \rightarrow R$ μια συνάρτηση για την οποία ισχύει $f(s) < f(s')$ $F(s) > F(s')$ για κάθε ss' . Η συνάρτηση αυτή αποκαλείται fitness function

Η εξάτμιση της φερομόνης είναι απαραίτητη για να αποφεύγεται η γρήγορη σύγκλιση του αλγορίθμου και η ενίσχυση των αρχικών τυχαίων διακυμάνσεων. Ο ακριβής τρόπος ανανέωσης της φερομόνης, δηλαδή η επιλογή του S_{upd} και της F είναι χαρακτηριστικό της κάθε υλοποίησης του αλγορίθμου.

2.7.2 Εξέλιξη και Παραλλαγές του αλγόριθμου ACO

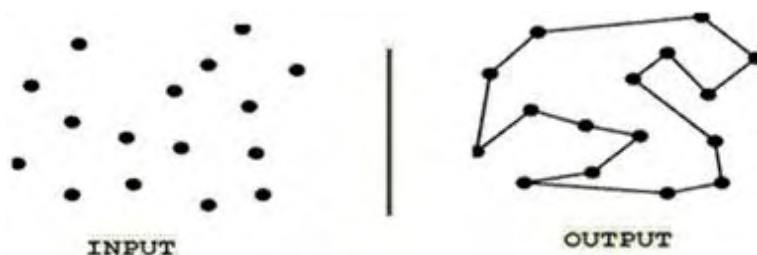
2.7.2.1 Ο αλγόριθμος Ant System (AS)

Όπως ήδη έχουμε αναφέρει και στην εισαγωγή του κεφαλαίου ο πρώτος αλγόριθμος που αναπτύχθηκε με βάση το μοντέλο ACO είναι ο *Ant System* (AS) του με στόχο την επίλυση του προβλήματος του Περιπλανώμενου Πωλητή-TSP καθώς το πρόβλημα αυτό είναι συναφές με τη πραγματική συμπεριφορά των μυρμηγκιών κατά την ανίχνευση και συγκομιδή τροφής, δηλαδή της εύρεσης της γρηγορότερης διαδρομής

Για το λόγο αυτό ο αλγόριθμος AS θα παρουσιασθεί στη μορφή για την επίλυση του TSP. Πιο συγκεκριμένα, στο TSP το ζητούμενο είναι η εύρεση της συντομότερης διαδρομής που συνδέει n πόλεις. Οι περιορισμοί που τίθενται είναι:

- κάθε πόλη πρέπει να περιλαμβάνεται μια φορά στη διαδρομή
- η διαδρομή πρέπει να είναι κλειστή, δηλαδή ο «πωλητής» πρέπει να επιστρέψει στην πόλη από όπου ξεκίνησε.

Οι αποστάσεις μεταξύ των πόλεων δεν είναι απαραίτητο να είναι συμμετρικές καθώς η απόσταση από την πόλη i στη j μπορεί να είναι διαφορετική από την απόσταση από την j πόλη στην i (τότε ορίζεται το ATSP: *Asymmetric Traveling Salesman Problem*).



Εικόνα 2.4 Λύση TSP

Η κάθε λύση για το TSP σχηματίζεται με τη διαδοχική μετάβαση των “ψηφιακών” μυρμηγκιών από τη μια πόλη στην άλλη με κάποια πιθανότητα. Το “ταξίδι” ολοκληρώνεται όταν το “ψηφιακό” μυρμήγκι επιστρέψει στην αρχική πόλη. Τότε βαθμολογείται η λύση που έχει επιτύχει με κριτήριο το συνολικό μήκος της διαδρομής και προστίθεται η ανάλογη φερομόνη στο μονοπάτι που ακολούθησε. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε άτομο της αποικίας μέχρι να συμπληρωθεί ο ζητούμενος αριθμός επαναλήψεων.

Ο αλγόριθμος σχηματίζεται με την παρακάτω διαδικασία. Έστω m ο αρχικός πληθυσμός των “ψηφιακών” μυρμηγκιών. Μετά από πειράματα, ο Dorigo διαπίστωσε ότι $m=n$, δηλαδή ο αρχικός πληθυσμός πρέπει να είναι ίσος με τον αριθμό των πόλεων του προβλήματος. Αν είναι πολύ μεγάλος ο πληθυσμός της αποικίας τότε πραγματοποιείται σύγκλιση σε τοπικά βέλτιστα, ενώ αν είναι πολύ μικρός δεν λειτουργεί η στιγμεργετική επικοινωνία μεταξύ των μυρμηγκιών. Τα μυρμήγκια τοποθετούνται είτε τυχαία στις πόλεις είτε το καθένα σε μια διαφορετική πόλη ως αφετηρία της διαδρομής τους. Η πόλη αυτή καταγράφεται ως νούμερο “ένα” στη λίστα της μνήμης (*tabu list*) κάθε μυρμηγκιού. Η *tabu list* είναι μια λίστα που περιέχει τις πόλεις που έχει ήδη επισκεφθεί το μυρμήγκι έτσι ώστε αυτές να αποκλειστούν από τις επόμενες δυνατές επιλογές του.

Κάθε μυρμήγκι k κατά την επανάληψη t που βρίσκεται στην πόλη i επιλέγει την επόμενη πόλη j που θα επισκεφθεί με βάση τον τυχαίο - αναλογικό κανόνα μετάβασης (*randomproportional transition rule*):

$$p_{ij}^k(t) = \begin{cases} \frac{|\tau_{ij}(t)|^\alpha |n_{ij}(t)|^\beta}{\sum_{j \in J_i^k} |\tau_{ij}(t)|^\alpha |n_{ij}(t)|^\beta} , & \text{αν } j \notin J_i^k \\ 0 , & \text{αν } j \in J_i^k \end{cases}$$

Όπου:

- I_i^k είναι οι πόλεις που έχει ήδη επισκεφθεί το μυρμήγκι k όταν βρίσκεται στην πόλη i (*Tabu list*)
- $n_{ij}(t)$ είναι η ορατότητα (*visibility*) και ορίζεται ως το αντίστροφο της απόστασης μεταξύ των πόλεων i και j , δηλαδή $n_{ij} = 1/d_{ij}$. Η ορατότητα εκφράζει την ευριστική προτίμηση (*heuristic desirability*) ως επόμενου σταθμού της πόλης j όταν το μυρμήγκι βρίσκεται στην πόλη i και βασίζεται καθαρά σε τοπικές πληροφορίες. Προφανώς, η πληροφορία αυτή είναι αναλλοίωτη στο TSP καθώς δεν μεταβάλλεται κατά την επίλυση του προβλήματος, δηλαδή $n_{ij}(t) = n_{ij}$
- $\tau_{ij}(t)$ έχει οριστεί στην εξίσωση (2) της §2.7.1
- α και β είναι παράμετροι που καθορίζονται από τον χρήστη και εξαρτώνται από το πρόβλημα προς επίλυση. Καθορίζουν το βαθμό που θα συνεισφέρει η τοπική γνώση (στο TSP η απόσταση μεταξύ των πόλεων) και η θετική ανάδραση της φερομόνης.

Αφού όλα τα μυρμήγκια ολοκληρώσουν τα «ταξίδια» τους κατά την επανάληψη t , τότε προστίθεται φερομόνη σε κάθε μονοπάτι ανάλογα με την επίδοση του κάθε μυρμηγκιού. Το ρόλο της F (*fitness function*) παίζει το αντίστροφο της συνολικής απόστασης του μονοπατιού πολλαπλασιασμένο με μια σταθερά Q η οποία επιλέγεται από τον χρήστη.

Άρα οι νέες τιμές φερομόνης είναι οι εξής :

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

$$\text{Όπου } \Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & , \text{ αν } (i, j) \in T^k(t) \\ 0 & , \text{ αν } (i, j) \notin T^k(t) \end{cases}$$

με :

- $T^k(t)$: το ταξίδι του μυρμηγκιού k , δηλαδή η σειρά των πόλεων που επισκέφθηκε κατά την επανάληψη t
- $L^k(t)$: το μήκος του ταξιδιού $T^k(t)$

2.7.2.2 Elitist ant system

Ο Dorigo αναπαράγοντας και προσαρμόζοντας την έννοια του ελιτισμού όπως αυτή χρησιμοποιείται στους εξελικτικούς αλγορίθμους εισήγαγε την ιδέα των “ελίτ” μυρμηγκιών (*elitist ants*) στον αλγόριθμο AS δημιουργώντας τον αλγόριθμο ASe. Σε αυτή την περίπτωση ένας επιπλέον αριθμός μυρμηγκιών, έστω e , ενισχύει την βέλτιστη διαδρομή T_{best} που έχει βρεθεί έως και την τρέχουσα επανάληψη t με επιπλέον φερομόνη ίση με eQ / L_{best} .

Συνεπώς, στην παραπάνω εξίσωση ανανέωσης φερομόνης προστίθεται ο παράγοντας $\Delta \tau_{ij}^e(t)$ με :

$$\Delta \tau_{ij}^e(t) = \begin{cases} Q/L_{best}(t) & , \text{αν } (i,j) \in T_{best}(t) \\ 0 & , \text{αν } (i,j) \notin T_{best}(t) \end{cases}$$

και :

- ✓ $L_{best}(t)$ το μήκος της βέλτιστης διαδρομής στην επανάληψη t
- ✓ $T_{best}(t)$ η σειρά των πόλεων της βέλτιστης διαδρομής στην επανάληψη t

2.7.2.3 Ο αλγόριθμος Ant Colony System (ACS)

Ο αλγόριθμος αυτός εισήχθη από τον Dorigo και αποτελεί βελτίωση του AS. Μάλιστα, τα αποτελέσματα του ACS είναι σαφώς ανώτερα του AS για το TSP. Οι βασικές διαφορές των δύο αλγορίθμων μπορούν να συνοψιστούν στα εξής:

1. **Κανόνας μετάβασης (transition rule):** Τώρα πλέον, ένα μυρμήγκι k που βρίσκεται στην πόλη i επιλέγει την επόμενη πόλη j που θα επισκεφθεί με τον εξής κανόνα:

$$j = \max_{u \in J_i^k} \{ \tau_{iu}(t) [n_{iu}(t)]^\beta \} , \text{αν } q \leq q_0$$

$$j = J , \text{αν } q > q_0$$

Στις παραπάνω εξισώσεις :

- q : είναι μια τυχαία μεταβλητή ομοιόμορφα κατανομημένη στο διάστημα $[0,1]$
- q_0 : είναι μια καθοριζόμενη παράμετρος με τιμή στο διάστημα $[0,1]$
- J : είναι μια πόλη που επιλέγεται τυχαία σύμφωνα με την παρακάτω εξίσωση:

$$p_{ij}^k(t) = \frac{|\tau_{ij}(t)||n_{ij}(t)|^\beta}{\sum_{j \in J_i^k} |\tau_{ij}(t)||n_{ij}(t)|^\beta}, \text{ αν } j \notin J_i^k$$

Όταν $q \leq q_0$ τότε τα μυρμήγκια εκμεταλλεύονται (*exploit*) τον ήδη εξερευνημένο χώρο, δηλαδή επιλέγουν τις τοπικά βέλτιστες λύσεις χρησιμοποιώντας τη συσσωρευμένη εμπειρία (επίπεδα φερομόνης και ορατότητας). Προφανώς, η επιλογή των τοπικά βέλτιστων λύσεων δεν είναι σίγουρο πως οδηγεί και σε ολικό βέλτιστο.

Αντίθετα, για $q > q_0$ πραγματοποιείται επιπλέον εξερεύνηση του χώρου των λύσεων. Όπως στον AS, έτσι και εδώ, τα μυρμήγκια επιλέγουν την καλύτερη έως τώρα λύση με κάποια πιθανότητα, ενώ δεν αποκλείεται να «χαθούν» με ενδεχόμενο, αργότερα, να ανακαλύψουν ένα νέο ολικό βέλτιστο

2. **Ανανέωση της φερομόνης (pheromone update):** Η εναπόθεση φερομόνης γίνεται μόνο από το μυρμήγκι που μέχρι αυτή τη χρονική στιγμή έχει κάνει την καλύτερη διαδρομή (αντί από όλα τα μυρμήγκια όπως στον AS) σε μια προσπάθεια να κατευθυνθεί η εξερεύνηση του χώρου προς την περιοχή του ολικού βέλτιστου.

Συνεπώς, η σχέση που διέπει την εναπόθεση της φερομόνης μετασχηματίζεται στην εξής (κανόνας εναπόθεσης φερομόνης στη βέλτιστη διαδρομή) :

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \begin{cases} Q/L_{best}(t) & , \text{ αν } (i,j) \in T_{best}(t) \\ 0 & , \text{ αν } (i,j) \notin T_{best}(t) \end{cases}$$

με :

- ✓ $L_{best}(t)$ το μήκος της βέλτιστης διαδρομής στην επανάληψη t
- ✓ $T_{best}(t)$ η σειρά των πόλεων της βέλτιστης διαδρομής στην επανάληψη t

3. **Τοπική ανανέωση της φερομόνης (local pheromone update):** Για να αποφευχθεί ο εγκλωβισμός των μυρμηγκιών σε μια και μόνο διαδρομή, κάτι που θα έκανε τη χρήση των m μυρμηγκιών ατελέσφορη, πρέπει κάθε μυρμήγκι k που επισκέπτεται την πόλη j από την πόλη i , να αφαιρεί ένα ποσό φερομόνης από το μονοπάτι που συνδέει αυτές τις δύο πόλεις. Με αυτόν τον τρόπο, η εξερεύνηση για νέες λύσεις κατευθύνεται μακριά από την καλύτερη

που έχει υπολογισθεί έως αυτή τη στιγμή. Η σχέση που ισχύει σε αυτή την περίπτωση είναι:

$$\tau_{ij}(t+1) \leftarrow (1 - \rho)\tau_{ij} + \rho\tau_0$$

όπου τ_0 η ποσότητα της φερομόνης με την οποία γίνεται η αρχικοποίηση.

Έχει βρεθεί πειραματικά ότι: $\tau_0 = (n Lnn)^{-1}$ με n τον αριθμό των πόλεων και Lnn το μήκος μιας διαδρομής που δημιουργείται με τον αλγόριθμο *Nearest Neighbor* (άπληστος αλγόριθμος λύσης του TSP ο οποίος επιλέγει ως επόμενο σταθμό την πιο κοντινή πόλη).

4. **Χρήση λίστας υποψηφίων πόλεων (candidate list):** Στα προβλήματα με μεγάλο αριθμό πόλεων – και ειδικά σε προβλήματα ATSP – είναι χρονοβόρο να εξετάζονται όλες οι πόλεις. Για το λόγο αυτό η λίστα αυτή περιέχει ένα αριθμό από τις πιο «επιθυμητές», δηλαδή κοντινές, πόλεις ταξινομημένες από την κοντινότερη προς την πιο μακρινή. Μόνο όταν οι πόλεις που βρίσκονται στη λίστα των υποψηφίων πόλεων ανήκουν και στην *Tabu list* είναι δυνατόν τα μυρμήγκια να κατευθυνθούν σε άλλες πόλεις

5. Άλλες διαφοροποιήσεις:

- a. ανανέωση της φερομόνης από τα δύο μυρμήγκια με τις καλύτερες διαδρομές
- b. αφαίρεση φερομόνης από τις κακές διαδρομές
- c. χρήση αλγορίθμων για τοπική αναζήτηση λύσεων, πχ. 2-opt, 3-opt, Lin-Kernighan.

2.7.2.4 Ο αλγόριθμος MAX-MIN ANT SYSTEM (MMAS)

Ο πιο πρόσφατος αλγόριθμος που έχει βελτιώσει σημαντικά τον AS και μάλιστα δίνει και καλύτερα αποτελέσματα από τον ACS στο TSP είναι ο MMAS του Stützle. Επιγραμματικά οι διαφοροποιήσεις από τον AS είναι οι εξής:

- ✓ Σε κάθε επανάληψη εναποτίθεται φερομόνη μόνο από ένα μυρμήγκι : Το μυρμήγκι που εναποθέτει τη φερομόνη είναι αυτό που έχει βρει την ολική βέλτιστη λύση (*global best ant*) ή αυτό που βρίσκει την καλύτερη λύση σε κάθε επανάληψη (*iteration best ant*). Προφανώς είναι δυνατή η χρήση διαφόρων συνδυασμών μεταξύ αυτών των δύο. Για παράδειγμα, στην αρχή μπορεί να χρησιμοποιείται η καλύτερη λύση κάθε επανάληψης για να εξερευνηθεί (*exploration*) όσο το δυνατόν καλύτερα όλος ο χώρος των λύσεων και καθώς αυξάνεται ο αριθμός των επαναλήψεων χρησιμοποιείται όλο και περισσότερο η ολική βέλτιστη λύση έτσι ώστε να γίνει εκμετάλλευση (*exploitation*) της περιοχής της λύσης αυτής.

- ✓ **Χρήση ορίων στα επίπεδα της φερομόνης:** Στον MMAS χρησιμοποιεί τόσο άνω όριο τ_{max} όσο και κάτω τ_{min} για τον περιορισμό της φερομόνης. Ο σκοπός του άνω ορίου είναι να μην επιτρέψει να εμφανιστεί στασιμότητα (*stagnation*) κατά την επίλυση του προβλήματος καθώς κανένα μονοπάτι δεν θα μπορεί να συγκεντρώσει τόση φερομόνη – και κατά συνέπεια να παρουσιάσει μεγάλη πιθανότητα επιλογής – ώστε να έλκει πάντοτε όλα τα μυρμηγκία. Επιπλέον, το κάτω όριο της φερομόνης διασφαλίζει το ότι καμιά διαδρομή δεν θα έχει μηδενική ή περίπου μηδενική πιθανότητα εκλογής.

Τα άνω και κάτω όρια της φερομόνης ορίζονται με τις παρακάτω σχέσεις:

$$\tau_{max} = \frac{1}{\rho \cdot L_{opt}}$$
$$\tau_{min} = \frac{\tau_{max} (1 - \sqrt[n]{p_{best}})}{(n/2 - 1) \sqrt[n]{p_{best}}}$$

όπου:

- L_{opt} : το μήκος της πραγματικής / ολικής βέλτιστης λύσης. Επειδή το μήκος αυτό δεν είναι γνωστό εκ των προτέρων χρησιμοποιούμε στη θέση του το $L_{best}(t)$. Συνεπώς, το τ_{max} είναι δυναμικά μεταβαλλόμενο.
 - p_{best} : η πιθανότητα δημιουργίας της ολικά βέλτιστης λύσης. Η παράμετρος αυτή καθορίζεται από το χρήστη. Σημειώνεται ότι όταν $p_{best} = 1$ τότε $\tau_{min} = 0$. Επίσης, αν η τιμή του p_{best} είναι πολύ μικρή τότε υπάρχει πιθανότητα να ισχύει $\tau_{min} > \tau_{max}$. Στην περίπτωση αυτή, θέτουμε $\tau_{min} = \tau_{max}$ οπότε και ο αλγόριθμος χρησιμοποιεί μόνο την ευριστική πληροφορία (ορατότητα) για τη λύση του προβλήματος.
 - n : ο αριθμός των πόλεων.
-
- ✓ **Αρχικοποίηση της φερομόνης στην τιμή τ_{max} , δηλαδή $\tau_o = \tau_{max}$:** Η διαφοροποίηση αυτή βοηθάει στην καλύτερη εξερεύνηση (*exploration*) του χώρου των λύσεων στις πρώτες επαναλήψεις του αλγορίθμου, καθώς η σχετική διαφορά μεταξύ των επιπέδων της φερομόνης των καλών και των κακών λύσεων είναι μικρή. Στο γεγονός αυτό συνεισφέρουν και τα όρια τ_{max} και τ_{min} . Αντιθέτως, όταν προστίθεται φερομόνη σε ένα αρχικά ομοιόμορφο χώρο με πολύ χαμηλά επίπεδα φερομόνης, η σχετική διαφορά των επιπέδων της φερομόνης των καλών και των κακών λύσεων είναι πολύ μεγάλη. Με αυτό τον τρόπο, ο αλγόριθμος οδηγείται στην εκμετάλλευση (*exploitation*) του χώρου των καλών λύσεων αγνοώντας, κατά κάποιον τρόπο, την εξερεύνηση (*exploration*) όλου του χώρου των λύσεων.

- ✓ **Ομαλοποίηση της φερομόνης (smoothing of the pheromone trail):** Όταν ο αλγόριθμος έχει συγκλίνει ή έχει φτάσει κοντά στη σύγκλιση τότε επιχειρείται ομαλοποίηση των επιπέδων της φερομόνης. Ο μηχανισμός αυτός πραγματοποιεί αύξηση των επιπέδων της φερομόνης ανάλογα με τη διαφορά τους από το τ_{max} έτσι ώστε να αυξηθεί η πιθανότητα επιλογής των μονοπατιών με χαμηλά επίπεδα φερομόνης. Η σύγκλιση (*convergence*) στον MMAS ορίζεται ως η κατάσταση στην οποία σε κάθε πόλη ένα από τα μονοπάτια που την συνδέουν με τις άλλες πόλεις έχει τιμή φερομόνης ίση με τ_{max} , ενώ όλα τα υπόλοιπα έχουν τιμή ίση με τ_{min} .

Η σχέση που υλοποιεί την παραπάνω διαδικασία έχει ως εξής:

$$\tau_{ij} \leftarrow \tau_{ij} + \delta (\tau_{max} - \tau_{ij})$$

όπου το δ είναι μια παράμετρος καθοριζόμενη από το χρήστη με $0 \leq \delta \leq 1$. Για $\delta=1$ έχουμε επαναρχικοποίηση των επιπέδων της φερομόνης, ενώ για $\delta=0$ απενεργοποιείται ο μηχανισμός αυτός.

Η διαδικασία αυτή χρησιμοποιείται κυρίως σε εκτελέσεις του αλγορίθμου με πολύ μεγάλο αριθμό επαναλήψεων έτσι ώστε να εξερευνάται καλύτερα ο χώρος των λύσεων. Επίσης, ο μηχανισμός αυτός κάνει πιο ανθεκτικό τον αλγόριθμο στις διάφορες τιμές του τ_{min} .

2.8 Αποτελέσματα - Συμπεράσματα

2.8.1 Σύγκριση των αλγορίθμων ACO

Ο Stützle έκανε μια σειρά πειραμάτων με διάφορες εκδόσεις του ACO πάνω σε μια σειρά χαρακτηριστικών TSP προβλημάτων. Τα αποτελέσματα είναι τα εξής:

Πρόβλημα	Opt	MMAS +pts	MMAS	ACS	ACS rank	ACS rank +ppts	ACSe	ACS +pts	AS
Eil51	426	427,1	427,6	428,1	434,5	428,8	428,3	427,4	437,3
kroA100	21182	21291	21320	21420	21746	21394	21522	21431	22471
D198	15780	15956	15972	16054	16199	16025	16205	16140	16702
Ry48p	14422	14523	14553	14565	14511	14644	14685	14657	15296
Ft70	38673	38922	39040	39099	39410	39199	39261	39161	19596
Ftv170	2755	2817	2828	2826	2854	2915	2952	2908	3154
Kro124p	3620	36573	36773	36857	36973	37218	37510	37417	38773

Πίνακας 2.1 Πειράματα σύγκρισης αλγορίθμων

Στον παραπάνω πίνακα:

1. Η πρώτη στήλη είναι τα προβλήματα, τα οποία χρησιμοποιούνται για να ελέγξουν την αποδοτικότητα αλγορίθμων.
2. Η δεύτερη στήλη είναι το μήκος της βέλτιστης διαδρομής
3. *+pts* : ο αλγόριθμος με την λειτουργία ομαλοποίησης φερομόνης ($\delta = 0.5$)
4. *ASrank* ένας παρόμοιος αλγόριθμος με αυτούς που αναλύθηκαν όπου ένας σταθερός αριθμός απ' τα καλύτερα μυρμηγκία εναποθέτει φερομόνη
5. *ASe* είναι ο αλγόριθμος AS με ελιτισμό, όπως περιγράφηκε παραπάνω
6. Τα αποτελέσματα αποτελούν μέσω όρο της βέλτιστης λύσης των αλγορίθμων.
7. Τα προβλήματα είναι TSP και ATSP.

Από τα αποτελέσματα είναι εμφανής η υπεροχή του MMAS έναντι των άλλων αλγορίθμων και η αδυναμία του απλού AS να προσεγγίσει την βέλτιστη λύση σε μερικά προβλήματα.

2.8.2 Σύγκριση με άλλους γενετικούς αλγορίθμους

Σε μια σύγκριση του ACS με άλλους αλγορίθμους σε συμμετρικό TSP τα αποτελέσματα είναι τα εξής:

Όνομα Προβλήματος	ACS	SA	SOM
City set 1	5,88	5,88	6,06
City set 2	6,05	6,01	6,25
City set 3	5,58	5,65	5,83
City set 4	5,74	5,81	5,87
City set 5	6,18	6,33	6,70

Πίνακας 2.2 Σύγκριση με αλγορίθμων με τον ACS

Τα αποτελέσματα είναι από δοκιμή από Dorigo και Gambardella (1997) σε πρώιμη ακόμα μορφή του αλγορίθμου χωρίς να χρησιμοποιείται ο ανώτερος αλγόριθμος MMAS. Δε χρησιμοποιείται επίσης, κανένα *local search* στη λύση για βελτίωση απόδοσης. Τα προβλήματα είναι τυχαία συμμετρικά TSP 50 πόλεων και τα αποτελέσματα είναι ο μέσος όρος της βέλτιστης λύσης από 25 επαναλήψεις. Η απόδοση του αλγορίθμου είναι εντυπωσιακή καθώς σχεδόν σε κάθε πρόβλημα επιτυγχάνει να ξεπεράσει τους άλλους αλγορίθμους και δείχνει την έφεσή του σε τέτοιου είδους συνδυαστικά προβλήματα.

2.8.3 Συμπεράσματα

Από τα πρώτα χρόνια της δημιουργίας του οι Ant Colony Optimization αλγόριθμοι έδειξαν την δύναμή τους και ανταγωνίστηκαν τους υπάρχοντες αλγόριθμους. Επίσης, είναι εμφανές ότι υπάρχει χώρος για βελτίωση του αλγορίθμου και πολλές προοπτικές να εξελιχθεί. Βασικές διευθύνσεις στις οποίες κινούνται οι μελετητές είναι :

1. *Δυναμικά συνδυαστικά προβλήματα*, στα οποία ο χώρος αναζήτησης αλλάζει με τον χρόνο και η ποιότητα των λύσεων που βρέθηκε μπορεί να μεταβληθεί, με βασικό παράδειγμα το routing στο χώρο των τηλεπικοινωνιών.
2. *Στοχαστικά συνδυαστικά προβλήματα*, στα οποία κάποιες μεταβλητές έχουν στοχαστική φύση.
3. *Βελτιστοποίηση προβλημάτων πολλαπλών στόχων*.
4. *Παράλληλη υλοποίηση*. Ο αλγόριθμος λόγω της φύσης του ευνοεί την παράλληλη υλοποίηση μοιράζοντας τα μυρμήγκια σε έναν αριθμό επεξεργαστών. Ο στόχος των μελετητών είναι να βρεθεί ο βέλτιστος αριθμός επεξεργαστών, μυρμηγκιών ανά επεξεργαστή αλλά και ο βαθμός και ο τρόπος επικοινωνίας ανάμεσα στους πληθυσμούς τους.
5. *Συνεχή ή μεικτά προβλήματα*. Ο αλγόριθμος είναι σχεδιασμένος για διακριτές μεταβλητές, οπότε μια απλή λύση θα ήταν να διακριτοποιηθεί η συνεχής μεταβλητή κάτι που σε μεγάλα προβλήματα δεν είναι εφικτή λύση. Άρα απαιτείται ειδική μελέτη αυτών των προβλημάτων.



Εικόνα 2.5 Μυρμήγκια σε σειρά

Κεφάλαιο 3ο

3 Κατασκευή προσομοιωτή αποικίας μυρμηγκιών

3.1 Εισαγωγή

Στα πλαίσια εκπόνησης της διπλωματικής εργασίας κατασκευάστηκε ένας προσομοιωτής έχοντας ως πρότυπο τον τρόπο που αναπτύσσονται οι αποικίες των μυρμηγκιών. Στην συνέχεια, ο προσομοιωτής επεκτάθηκε σε networks στην προσπάθεια να δείξουμε πως οι αλγόριθμοι ACO, μπορούν να προστατεύσουν και να ασφαλίσουν από κακόβουλες επιθέσεις υπολογιστικά δίκτυα. Τέλος, δημιουργήθηκαν δύο σενάρια εκτέλεσης με σκοπό την καλύτερη κατανόηση του αλγορίθμου.

Η κατασκευή του Ant Simulator έγινε με την βοήθεια του προγραμματιστικού εργαλείου Netbeans 7.0.1. Ο κώδικας γράφτηκε σε γλώσσα προγραμματισμού Java. Για την δημιουργία του γραφικού περιβάλλοντος του Simulator κατασκευάστηκε ένα GUI σε java.swing. Επίσης για το διάγραμμα κλάσεων UML που παρουσιάζεται πιο κάτω, χρησιμοποιήθηκε η trial έκδοση του προγράμματος Visual Paradigm for UML 8.0 Professional Edition. Τέλος, είναι σημαντικό να αναφέρουμε και την χρήση της JDK 1.6 από το Netbeans.

3.2 Γενική Περιγραφή συστήματος

Ο προσομοιωτής αποτελείται από 11 κλάσεις, οι συσχετίσεις των οποίων φαίνονται στο παρακάτω UML διάγραμμα κλάσεων. Επιγραμματικά αναφέρουμε τις λειτουργίες τους :

1. **MainWindow** : Πρόκειται για ένα εισαγωγικό JFrame στο οποίο αναφέρονται οι λειτουργίες που θα παρουσιαστούν στην συνέχεια. Μέσω αυτής της κλάσης γίνεται η εκκίνηση της κύριας εφαρμογής
2. **AntsApplication** : Κλάση για την κατασκευή του Ant application frame. Σε αυτή την κλάση κατασκευάζεται το παράθυρο της εφαρμογής μας στο οποίο θα απεικονιστεί η εκτέλεση του αλγορίθμου. Περιέχει τον σχεδιασμό και την κατασκευή των JMenu των διαφορετικών υποπεριπτώσεων που εξετάζουμε
3. **AntMovement** : Κλάση για την κίνηση του μυρμηγκιού. Χρήση για την αναγνώριση του στόχου GOAL που θέλει να πετύχει. Εμπεριέχει την συνάρτηση die() μέσω της οποίας ουσιαστικά ορίζεται η διάρκεια ζωής του μυρμηγκιού την οποία καλούμε δύο φορές στον κώδικα μας. Πρώτον όταν το μυρμηγκί βρει όλες τις τροφές και γυρίζει στην φωλιά καλείται για να πεθάνει

και δεύτερον, ανά τυχαία (random) χρονικά διαστήματα προκειμένου να προσομοιώσουμε αστάθμητους εξωτερικούς παράγοντες πχ. πλημμύρα, βροχή κ.α. Επίσης εμπεριέχει την μέθοδο step() μέσω της οποίας το μυρμήγκι αποφασίζει πως επιλέγει να κινηθεί στο επόμενο βήμα. Ανανεώνει τον πίνακα των φερομονών της πηγής με την τρέχουσα μεγαλύτερη τιμή από όλα τα γειτονικά κελιά βάσει της :

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \sum_{s \in S_{upd}} F(s)$$

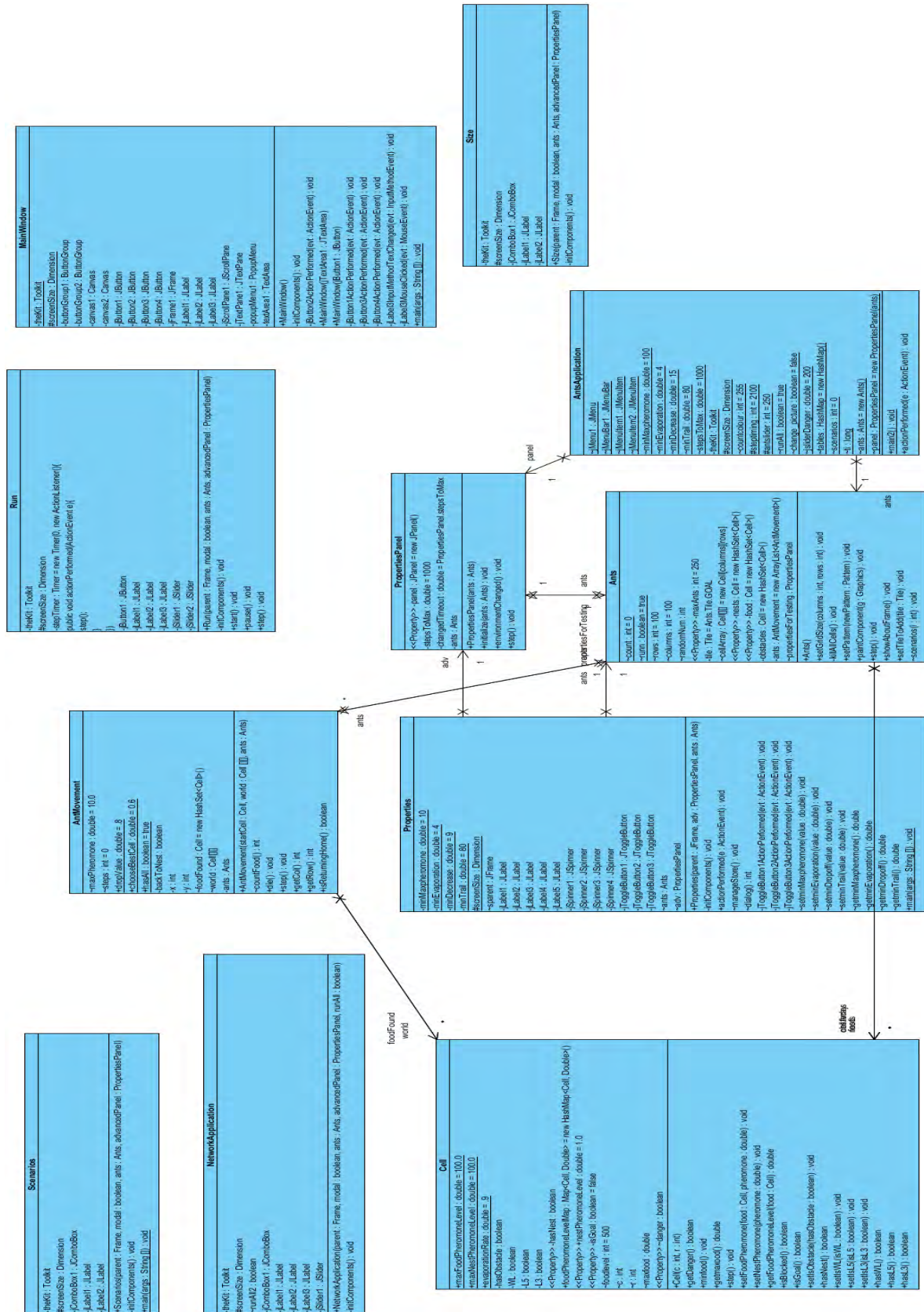
Αν βρει τροφή στο κελί που βρίσκεται παίρνει ένα μέρος της (μείωση κατά 0.1) και ανανεώνει τον πίνακα φερομονών τροφής . Όταν βρει όλες τις τροφές (λύση TSP) έχει επιτύχει τον στόχο του GOAL οπότε γυρίζει στην φωλιά. Τα βήματα αυτά πραγματοποιούνται σε επανάληψη μέχρι να καταναλωθούν όλες οι τροφές.

4. **Cell:** αυτή η κλάση υπολογίζει την πιθανότητα :

$$p_{ij}^k(\mathbf{t}) = \begin{cases} \frac{|\tau_{ij}(t)|^\alpha |n_{ij}(t)|^\beta}{\sum_{j \in J_i^k} |\tau_{ij}(t)|^\alpha |n_{ij}(t)|^\beta} , & \text{αν } j \notin J_i^k \\ 0 , & \text{αν } j \in J_i^k \end{cases} = \mathbf{1 - P_B}$$

που θα χρησιμοποιηθεί από την AntMovement για να ανανεώσει όλους τους πίνακες φερομόνης

5. **Ants:** Κλάση για την ενεργοποίηση των κελιών του πλέγματος που δημιουργήσαμε. Αν είναι food τότε το κελί θέτει την μεταβλητή του goal = true, αν είναι obstacle θέτει τη μεταβλητή obstacle = true και αν είναι nest τη μεταβλητή nest = true. Στην συνέχεια όλος ο simulator ενημερώνεται και εμφανίζει τις αλλαγές
6. **NetworkApplication :** κλάση υπεύθυνη για την διαμόρφωση και την σωστή εμφάνιση του παράθυρου Network Security
7. **Properties:** Με την κλάση αυτή ο χρήστης έχει την δυνατότητα να αυξομειώσει βασικές μεταβλητές. Συγκεκριμένα, το μέγεθος της φερομόνης που εναποτίθεται σε κάθε κελί, είτε αυτή προέρχεται μετά την εύρεση τροφής είτε από την φωλιά, το πόσο γρήγορα εξατμίζεται η φερομόνη, πόσο αυστηρά ή πιο ελεύθερα ακολουθούν τα μυρμήγκια το μονοπάτι που έχει ήδη βρεθεί και τέλος, το πόσο γρήγορα εξατμίζεται η φερομόνη που αφήνουν τα μυρμήγκια για να βρουν το δρόμο της επιστροφής.

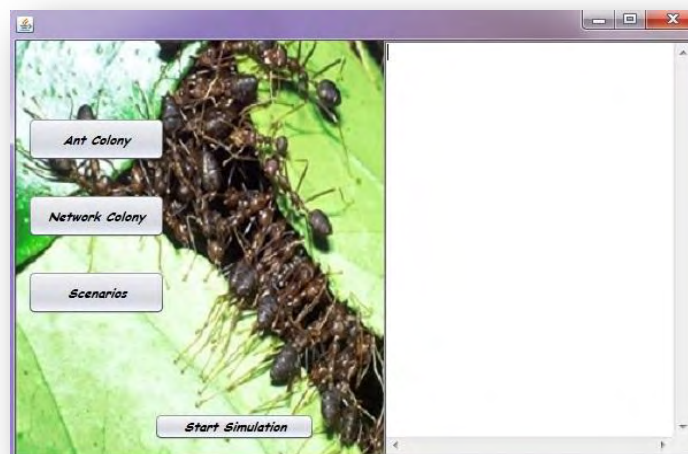


Εικόνα 3.1 Διάγραμμα κλάσεων UML

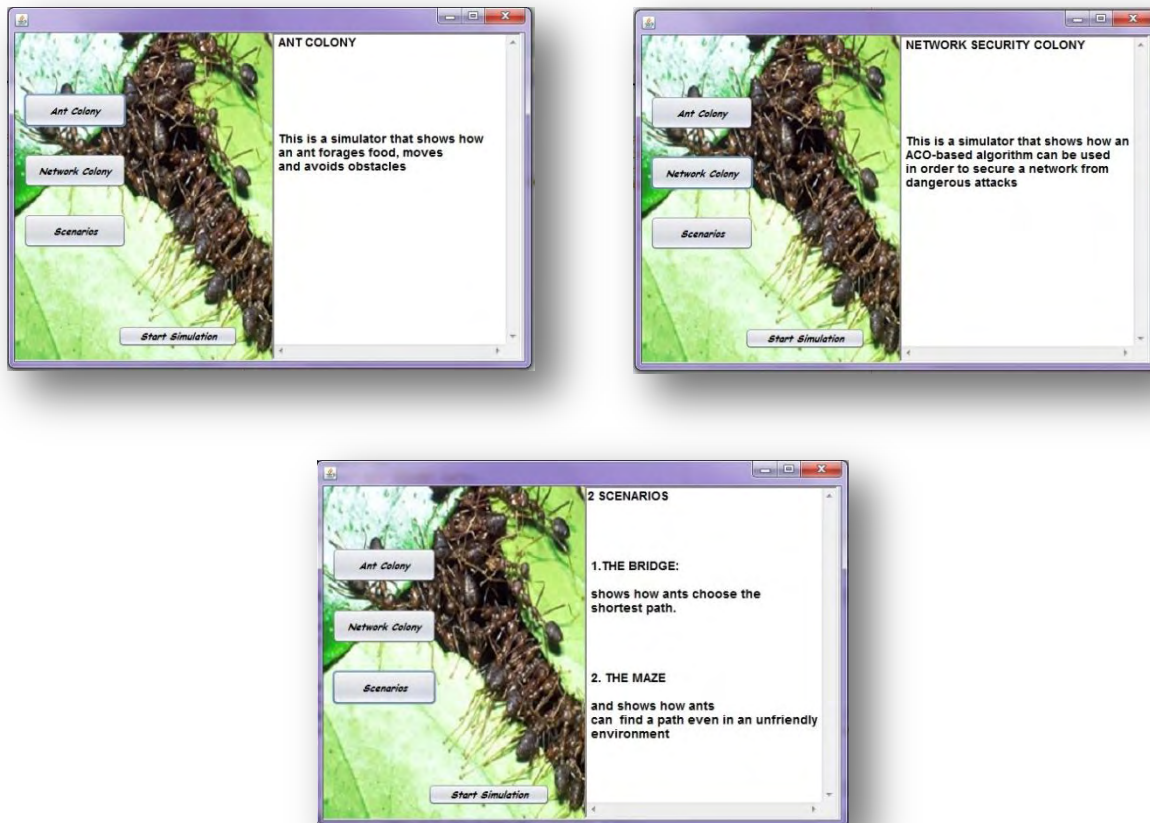
8. **PropertiesPanel** : κλάση που περιέχει την διαμόρφωση του JPanel για την εμφάνιση του παράθυρου μέσα από το οποίο μπορούμε να αυξομειώσουμε τις μεταβλητές που αναφέραμε την κλάση Properties
9. **Run** : κλάση μέσω της οποίας δημιουργείται το παράθυρο για την εκτέλεση του αλγορίθμου. Περιέχει δύο jSlider1 με τα οποία μπορούμε να αλλάξουμε τον αριθμό μυρμηγκιών που παράγει η φωλιά καθώς και την ταχύτητα εκτέλεσης. Τέλος έχουμε την δυνατότητα παύσης της εκτέλεσης
10. **Scenarios**: κλάση για την δημιουργία παράθυρου με την δυνατότητα επιλογής δύο σεναρίων εκτέλεσης
11. **Size**: κλάση για την επιλογή του μεγέθους του πλέγματος της εφαρμογής μας. Έχουμε την δυνατότητα να επιλέξουμε μεταξύ 50 x 50 ή 100 x 100 πλέγμα γραφήματος

3.3 Ανάλυση γραφικού περιβάλλοντος

Εκτελώντας την συνάρτηση MainWindow ανοίγει το κεντρικό μας applet το οποίο μας ενημερώνει τι θα εξετάσουμε στην συνέχεια Από αυτό το παράθυρο επιλέγοντας τα κουμπιά Ant Colony, Network Colony και Scenarios μπορούμε να μάθουμε κάποια πράγματα για τα διαφορετικά κομμάτια της προσομοίωσης που θα ακολουθήσουν.

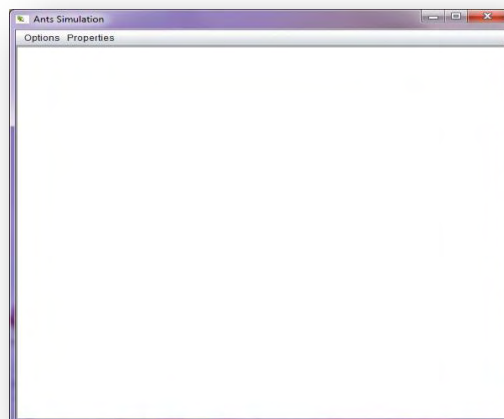


Εικόνα 3.2 Αρχικό παράθυρο εφαρμογής



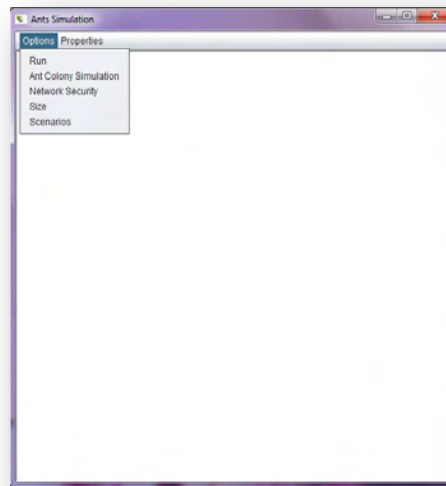
Εικόνα 3.3 Σύντομη εξήγηση των λειτουργιών

Τέλος πατώντας το κουμπί Start Simulation θα μεταφερθούμε σε ένα καινούριο applet μέσα από το οποίο θα παρακολουθήσουμε μια προσομοίωση του τρόπου με τον οποίο αναζητούν την τροφή τους τα μυρμήγκια, μια αναλογία του αλγορίθμου που μπορεί να χρησιμοποιηθεί για την ασφάλεια των δικτύων από εισβολείς, ενώ μπορεί να εκτελέσει δύο διαφορετικά σενάρια του αλγορίθμου για καλύτερη κατανόηση.



Εικόνα 3.4 Κεντρικό παράθυρο Ant Simulator

Στο μενού επιλογών που ανοίγει κάτω από το κουμπί Options μπορούμε να βρούμε τις λειτουργίες **Run, Ant Colony Simulation, Network Security, Size** και **Scenarios**



Εικόνα 3.5 Menu Options

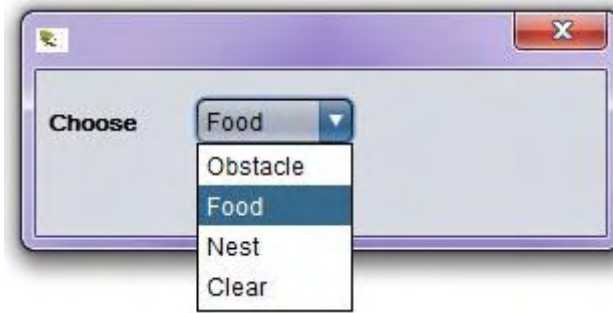
Run: με την επιλογή Run μπορούμε να τρέξουμε την εφαρμογή μας, να την κάνουμε να μπει σε κατάσταση παύσης (πάγωμα της κίνησης των μυρμηγκιών), ενώ μας δίνεται η δυνατότητα να αυξομειώσουμε τόσο την ταχύτητα κίνησης όσο και τον αριθμό των μυρμηγκιών.



Εικόνα 3.6 Επιλογές Run

Η μέγιστη τιμή του πληθυσμού των μυρμηγκιών που μπορεί να παράγει μια φωλιά είναι 250 μυρμηγκία ενώ η μέγιστη ταχύτητα έχει οριστεί σε 2100.

Ant Colony Simulation: Διαλέγοντας την Ant Colony Simulation μπορούμε να εισάγουμε σε όποια θέση του πλέγματος της εφαρμογής είτε τροφή (Food), είτε κάποιο εμπόδιο (Obstacle). Το εμπόδιο αυτό θα φράζει τον δρόμο του μυρμηγκιού και θα το αναγκάζει να βρει νέες εναλλακτικές διαδρομές. Τέλος έχουμε την επιλογή καθαρισμού (Clear) κάποιου αντικείμενου από αυτά που αναφέραμε και δεν χρειαζόμαστε πια στην εκτέλεσή μας.



Εικόνα 3.6 Επιλογές Ant Colony Simulation

Network Security: Από το μενού Options αν διαλέξουμε Network Security τότε θα μεταβούμε σε ένα περιβάλλον προσομοίωσης του αλγορίθμου, σε δίκτυα. Εδώ έχουμε τρία είδη δικτύων, ένα ασύρματο (Wireless) Lan αποτελούμενο από τέσσερις ασύρματες συσκευές, καθώς και δύο σταθερά δίκτυα Lan το ένα αποτελούμενο από πέντε υπολογιστές και το άλλο από τρεις. Επίσης το σύστημα διαθέτει μπάρα αυξομείωσης κινδύνου - Danger μέσω της οποίας ο χρήστης μπορεί να επιλέξει το μέγεθος επικινδυνότητας μιας επίθεσης, που δέχεται κάποιο από τα δίκτυα μας. Θέτουμε ένα άνω κατώφλι. Έτσι αν η τιμή του κινδύνου είναι κάτω από 50 τότε το σύστημά μας απλά το αγνοεί. Αν είναι πάνω από 50 τότε οι mobile agents τύπου μυρμηγκιών ανιχνεύουν τον κίνδυνο και ακολουθώντας αλγόριθμο ACO κινούνται για να τον καταπνίξουν.



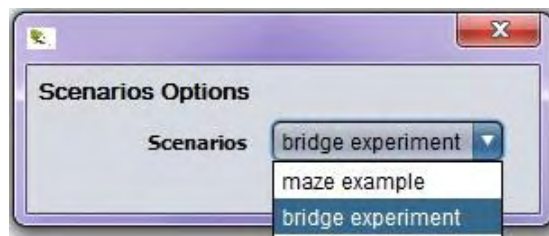
Εικόνα 3.7 Επιλογές Network Security

Size Options: Η επιλογή Size μας δίνει τη δυνατότητα να επιλέξουμε το μέγεθος του grid που θα δημιουργήσουμε. Η default επιλογή είναι η κατασκευή ενός πλέγματος 100 x 100. Με την επιλογή 50 μπορούμε ουσιαστικά να κάνουμε μεγέθυνση στο σύστημά μας ώστε να έχουμε καλύτερη οπτική της κίνησης των μυρμηγκιών. Χρήση αυτής της επιλογής θα αναλυθεί σε επόμενη ενότητα (Network Simulator).



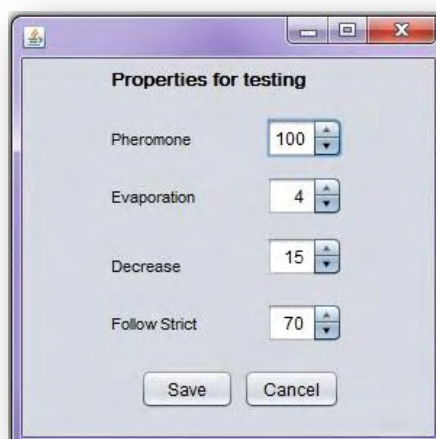
Εικόνα 3.8 Επιλογές Size

Scenarios Options: Άλλη μια λειτουργία του προσομοιωτή είναι αυτή της εκτέλεσης του πειράματος της διπλής γέφυρας και η επίδειξη του παραδείγματος του λαβυρίνθου.



Εικόνα 3.9 Επιλογές Scenarios Options

Properties: Τέλος κάτω από την επιλογή Properties μπορούμε να βρούμε τέσσερις μεταβλητές Pheromone, Evaporation, Follow Strict και DecreaseNP. Οι μεταβλητές αυτές, αναλύονται σε επόμενη ενότητα και χρησιμοποιούνται για πειραματισμό ώστε να κατανοήσουμε πως επηρεάζουν το περιβάλλον του πράκτορα Ant



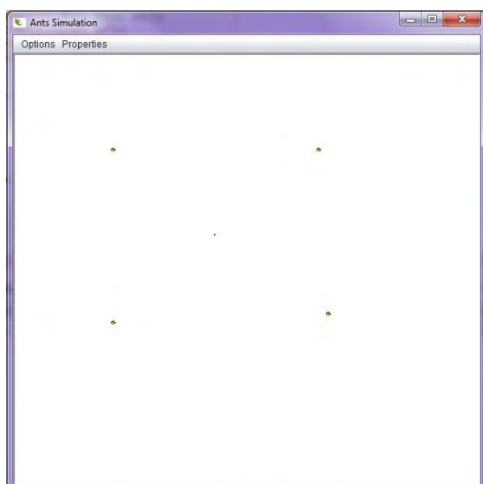
Εικόνα 3.10 Επιλογές Properties

3.3 Ant Colony Simulation

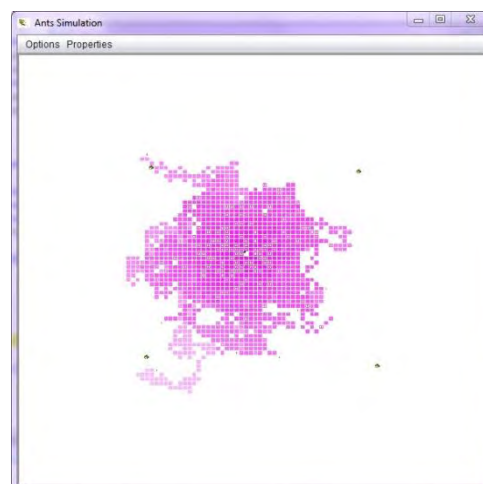
Στο αρχικό παράθυρο επιλέγουμε να εισάγουμε τέσσερις διαφορετικές τροφές καθώς και μία φωλιά. Η τροφή αποτελεί τον κύριο στόχο των μυρμηγκιών agent. Τα μυρμηγκία ξεκινούν από την φωλιά και κινούνται πιθανοτικά μέσα στο πλέγμα μας. Επίσης αφήνουν πίσω τους φερομόνη ροζ χρώματος, που θα τους οδηγήσει πίσω στην φωλιά αφού βρουν την τροφή. Στον προσομοιωτή μας έχουμε βάλει ως προϋπόθεση, ο αλγόριθμος να λύνει το πρόβλημα TSP. Άρα εδώ ο τελικός στόχος GOAL του agent μυρμηγκιού είναι όχι μόνο να βρει μία αλλά όλες τις τροφές και μετά να επιστρέψει στη φωλιά. Λόγω του μεγάλου αριθμού των μυρμηγκιών οι πιθανές διαδρομές φωλιά - τροφή είναι πολλές. Με την πάροδο του χρόνου όμως, επιλέγεται η συντομότερη μιας και η συγκέντρωση φερομόνης στο μονοπάτι αυτό θα είναι πιο μεγάλη καθώς θα ανανεώνεται συχνότερα.

3.3.1 Εύρεση βέλτιστου μονοπατιού

Στην συνέχεια ακολουθεί επίδειξη σε στιγμιότυπα προκειμένου να αναλύσουμε τον τρόπο που εξελίσσεται το σύστημα ACO. Τοποθετούμε σε 4 διαφορετικά κελιά του πλέγματός τροφή καθώς και μία φωλιά στο κέντρο.

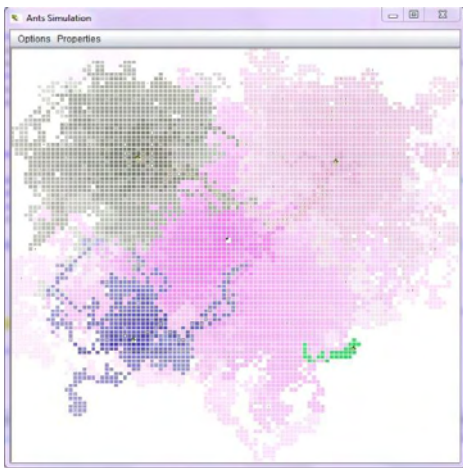


Εικόνα 3.11 Για $t = 0$ sec

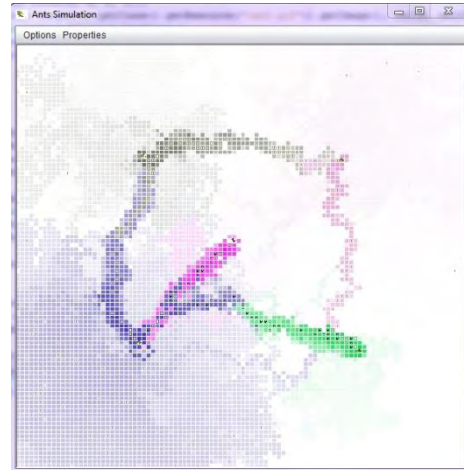


Εικόνα 3.12 Για $t = 10$ sec

Μετά από 10 sec από την εκκίνηση του προσομοιωτή βλέπουμε πως τα μυρμηγκία ξεχύνονται τυχαία προς όλες τις κατευθύνσεις αφήνοντάς του πίσω ροζ φερομόνη. Η φερομόνη της φωλιάς είναι πολύ σημαντική για την επιστροφή του μυρμηγκιού. Όταν ένα μυρμηγκί βρει και τις τέσσερις τροφές αλλάζει χρώμα και από καφέ γίνεται μαύρο για να δηλώσει πως τώρα γυρίζει στην φωλιά και ακολουθεί την ροζ φερομόνη

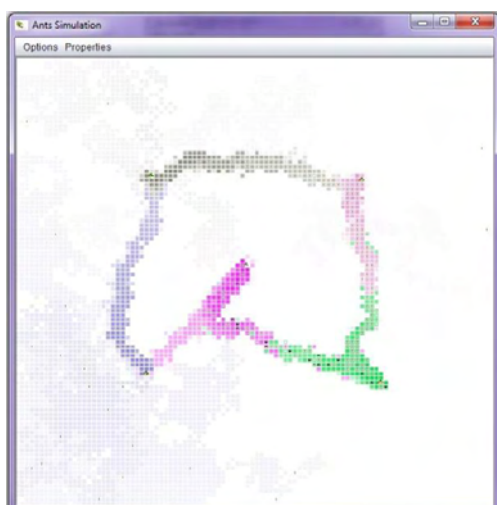


Εικόνα 3.12 Για $t = 40$ sec

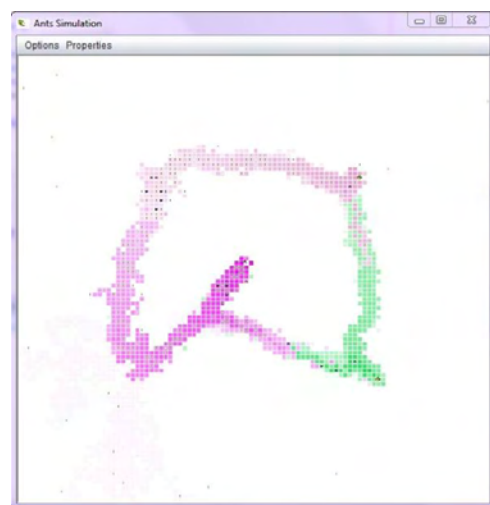


Εικόνα 3.13 Για $t = 70$ sec

Μόλις το μυρμήγκι βρεθεί πάνω σε κελί με τροφή παίρνει ένα κομμάτι της (την μειώνει κατά 0.1) και αρχίζει να αφήνει φερομόνη νέου χρώματος για να υποδηλώσει στα υπόλοιπα μυρμήγκια πως αν ακολουθήσουν το μονοπάτι που ακολούθησε το ίδιο σίγουρα θα βρουν τροφή (δημιουργία 4 διαφορετικών σημείων, μοβ σκούρο, κόκκινο σκούρο, ροζ σκούρο και γκρι). Μετά από χρόνο 40 sec παρατηρούμε πως τα μυρμήγκια έχουν βρει τις τέσσερις τροφές . Αφού ο αλγόριθμός μας θέλουμε να λύνει TSP τότε το μυρμήγκι δεν θα ακολουθήσει την ροζ φερομόνη πριν πάρει ένα κομμάτι τροφής από όλες τις τροφές. Τα μυρμήγκια στην αναζήτηση της τροφής θα βρουν πολλές εναλλακτικές επιλογές. Ωστόσο με την πάροδο του χρόνου τα πιο μακρινά μονοπάτια θα απορριφτούν και θα μείνει το βέλτιστο μονοπάτι που ενώνει στις 4 τροφές με την πηγή / φωλιά (Εικόνα 3.14). Όταν μια τροφή καταναλωθεί τότε η τροφή σβήνεται καθώς και το χρώμα της φερομόνης που έχουν αφήσει τα μυρμήγκια αφού την βρήκαν. Τα μυρμήγκια σταματούν να ακολουθούν την φερομόνη που οδηγεί σε αυτήν και επικεντρώνονται σε αυτές που έχουν παραμείνει στο περιβάλλον.

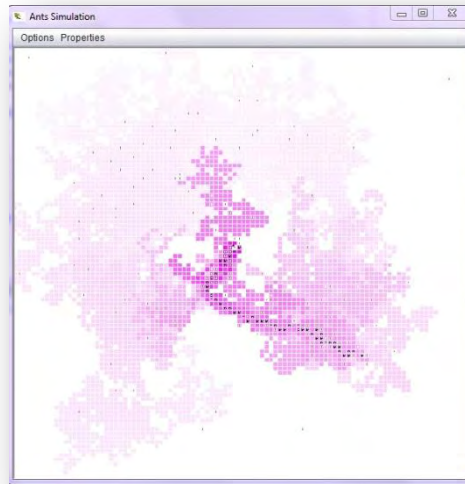


Εικόνα 3.14 Για $t = 140$ sec



Εικόνα 3.15 Για $t = 170$ sec

Γι'αυτό στην εικόνα 3.14 που έχουν εξαλειφτεί οι πάνω και οι κάτω αριστερά τροφές σιγά σιγά το μονοπάτι της φερομόνης εξασθενεί και παύει να ακολουθείτε. Όταν όλες οι τροφές καταναλωθούν τότε τα τελευταία μυρμηγκία επιστρέφουν. Όταν όλες οι τροφές έχουν φαγωθεί τότε όλα τα μυρμηγκία πεθαίνουν και γεννιέται καινούρια αποικία Εικόνα 3.16 . Επίσης στην προσπάθεια να προσομοιώσουμε τυχαίους εξωτερικούς παράγοντες , όπως για παράδειγμα μια πλημμύρα ή μια βροχή που σβήνει όλα τα ίχνη της φερομόνης εισάγαμε μια random μεταβλητή που προκαλεί τον ακαριαίο θάνατο όλων των μυρμηγκιών και την γέννηση νέων.



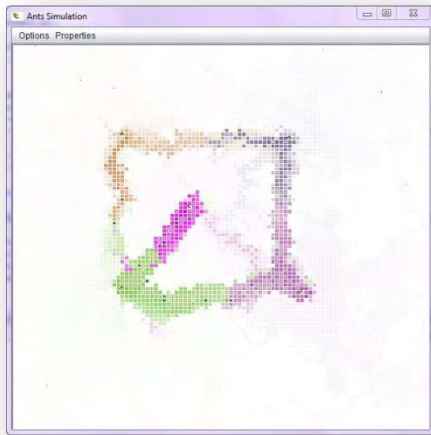
Εικόνα 3.16 Για $t = 210 \text{ sec}$

3.3.2 Προσαρμοστικότητα Ant Colony Simulator

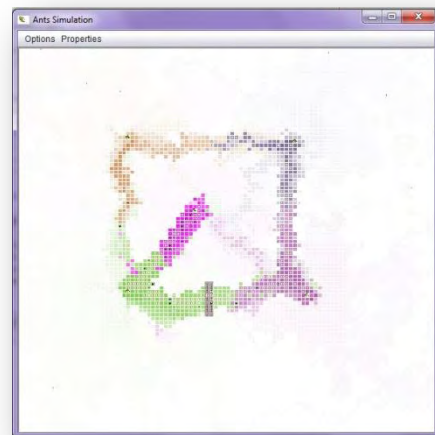
“Δεν είναι τα πιο δυνατά είδη που επιβιώνουν ή τα πιο έξυπνα, αλλά αυτά που ανταποκρίνονται καλύτερα στις αλλαγές”

Κάρολος Δαρβίνος

Πολύ σημαντικό στοιχείο του συστήματός μας είναι η ανθεκτικότητα που επιδεικνύει στις εξωτερικές παρεμβάσεις. Γι'αυτό το λόγο προσθέσαμε την μεταβλητή Obstacle η οποία υψώνεται σαν τοίχος και απενεργοποιεί το κελί στο οποίο τοποθετείται. Έτσι για παράδειγμα αν η αποικία μας έχει βρει ένα μονοπάτι από την φωλιά προς όλες τις τροφές και ξαφνικά βρεθεί κάποιο εμπόδιο τότε τα μυρμηγκία πρέπει να βρουνε εναλλακτική διαδρομή η οποία θα είναι (με την πάροδο του χρόνου) η νέα βέλτιστη.



Εικόνα 3.17 Μονοπάτι

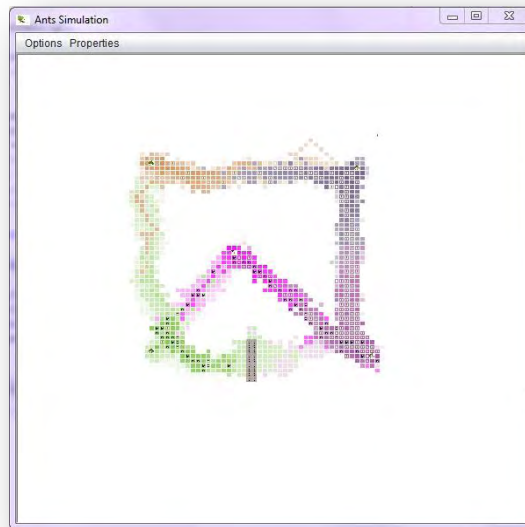


Εικόνα 3.17 Εισαγωγή εμποδίου



Εικόνα 3.18 Καταστροφή βέλτιστου μονοπατιού

Δημιουργούμε ένα μεγάλο τοίχο και κόβουμε την διαδρομή των μυρμηγκιών. Όπως φαίνεται και στην εικόνα με το ζουμ όλα τα μυρμηγκία χτυπάνε πάνω στον τοίχο. Το γεγονός αυτό θα διαρκέσει κάποια δευτερόλεπτα μέχρι να βρεθεί εντελώς τυχαία νέο μονοπάτι συντομότερο. Το νέο μονοπάτι μπορεί να είναι γύρω από τον τοίχο, ακολουθώντας το παλιό μονοπάτι και παρακάμπτοντας τον τοίχο, ή να είναι κάποιο άλλο μακριά από τον τοίχο το οποίο όμως να είναι το συντομότερο (όπως φαίνεται στην εικόνα)



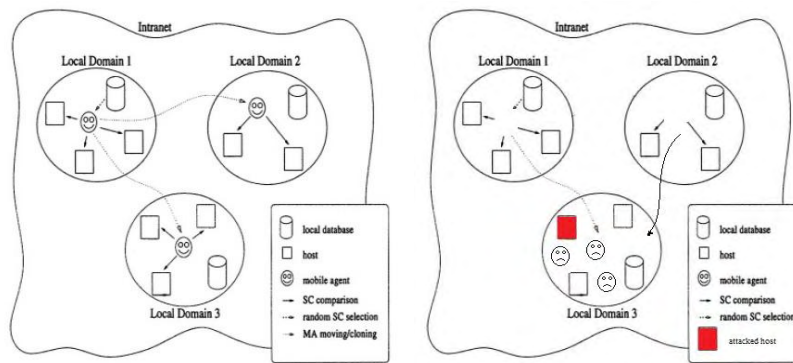
Εικόνα 3.18 Νέο βέλτιστο μονοπάτι

3.4 Network Security Simulation

Εμπνευσμένοι από το σύστημα IDRS (Intrusion Detection and Response System) των Noria Foukia και Salima Hassas θελήσαμε να φτιάξουμε μια προσομοίωση του αλγορίθμου ACO σε δίκτυα υπολογιστών.

Το σύστημα IDRS χωρίζεται σε δύο φάσεις . Η πρώτη είναι η ανίχνευση των πιθανών εισβολέων. Αυτό γίνεται μέσω ενός αλγορίθμου εμπνευσμένου από το ανοσοποιητικό σύστημα των ανθρώπων. Στην δεύτερη φάση, αφού δηλαδή έχουν αναγνωριστεί οι εισβολείς, αρχίζει η προσπάθεια καταστροφής των εισβολέων βάσει του αλγορίθμου ACO.

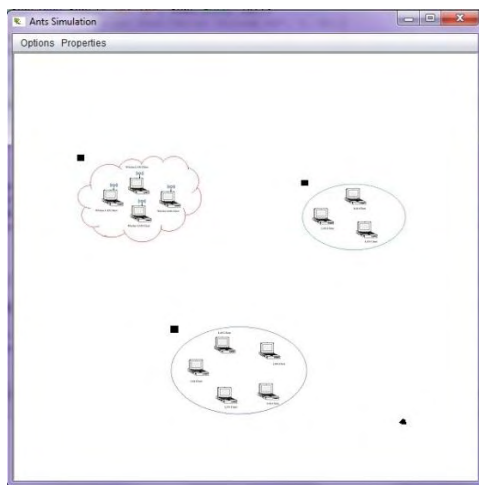
Η δική μας εφαρμογή, βασίζεται στην δεύτερη φάση, δηλαδή την επαναφορά του δικτύου μας σε κατάσταση πριν την επίθεση και την καταστολή των εισβολέων. Κάθε δίκτυο εμφανίζει μέχρι ένα ανεκτό σημείο - κατώφλι κάποια μορφή δυσλειτουργίας που ωστόσο δεν μπορεί να ληφθεί ως επίθεση. Έτσι εισάγαμε μια μεταβλητή Danger (επικινδυνότητα εισβολέα) η οποία λαμβάνει τιμές από 0 - 250. Αν η τιμή είναι κάτω από 50 τότε το σύστημα δεν θεωρεί υπολογίσιμο τον βαθμό επικινδυνότητας και δεν εκτελεί κάποια ενέργεια. Ωστόσο αν είναι πάνω από 50 τότε mobile agents με την μορφή μυρμηγκιών κινούνται για να καταπνίξουν τον/ τους εισβολείς.



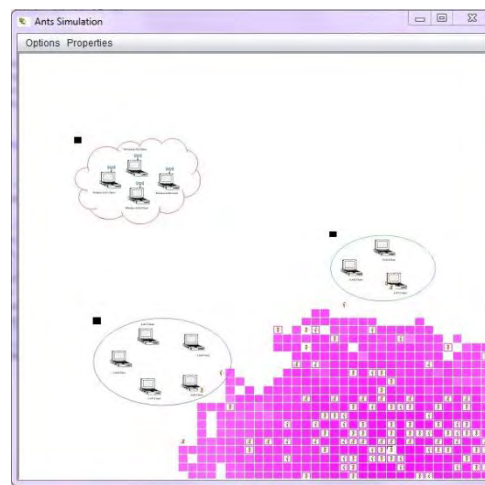
Εικόνα 3.19 α) Φάση αναγνώρισης β)Φάση καταστολής εισβολέων

3.4.1 Στιγμιότυπα εκτέλεσης Network Simulator

Αρχικά επιλέγουμε το μέγεθος το πλέγματος να είναι 50 x 50 (επιλογή από το μενού Size). Στη συνέχεια τοποθετούμε τρεις διαφορετικές τοπολογίες δικτύων. Ένα ασύρματο Wireless Lan αποτελούμενο από τέσσερις ασύρματες συσκευές, καθώς και δύο σταθερά δίκτυα αποτελούμενα από πέντε και από τρεις συσκευές αντίστοιχα. Τέλος, τοποθετούμε την φωλιά των mobile agents και επιλέγουμε να παράγει έως 150 άτομα (μυρμηγκία).



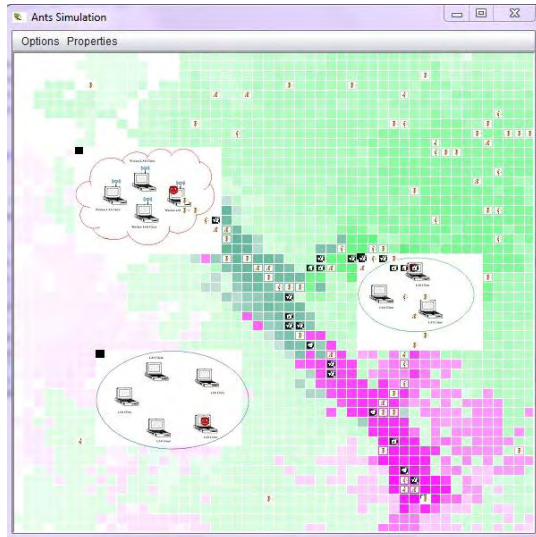
Εικόνα 3.20 Για $t=0$



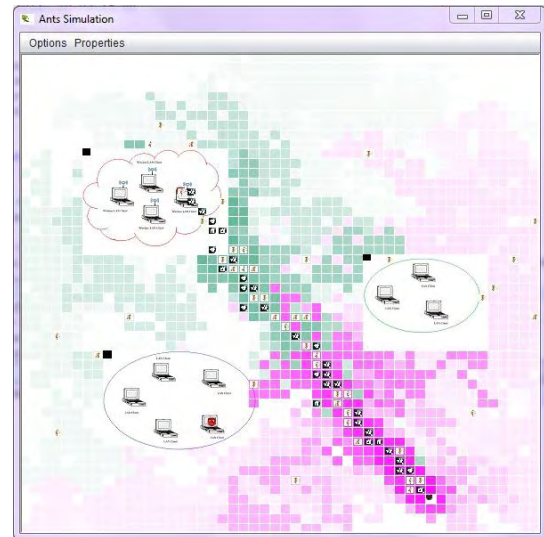
Εικόνα 3.21 Για $t=10$

Την χρονική στιγμή $t=30$ ξεκινάμε την επίθεση στο LAN με τους 3 υπολογιστές. Ο βαθμός επικινδυνότητας ορίζεται σε 120, έτσι θα αναγνωρισθεί η επίθεση και mobile agents θα κινηθούν προς το σημείο του εισβολέα. Την χρονική στιγμή $t = 35$ ξεκινάμε δεύτερη επίθεση στο Wireless δίκτυό μας με βαθμό επικινδυνότητας 200, ενώ για $t=40$ ξεκινάμε επίθεση στο LAN με τους 5 υπολογιστές με βαθμό επικινδυνότητας 30. Η δεύτερη επίθεση θα αναγνωρισθεί όπως η πρώτη και θα έχουμε την αντίστοιχη ενεργοποίηση των agents. Η τρίτη όμως, επειδή βρίσκεται μέσα στα ανεκτά όρια σφάλματος, απλά θα αγνοηθεί (Εικόνα 3.22). Έτσι οι mobile agents για να φέρουν τα δίκτυα στην αρχική τους

κατάσταση θα τρέξουν παρόμοιο αλγόριθμο με αυτόν που αναφέρθηκε στο Ant Colony Simulator λύνοντας παράλληλα και το πρόβλημα TSP (Εικόνα 3.23).



Εικόνα 3.22 Για $t=50$



Εικόνα 3.23 Για $t=150$

Τα καφέ χρώματος mobile agent - μυρμηγκία είναι αυτά που κινούνται μέσα στο χώρο πιθανοτικά και αναζητούν τους εισβολείς, ενώ τα μαύρα είναι αυτά που έχουν βρει όλους τους εισβολείς και γυρίζουν στην “φωλιά” προκειμένου να ολοκληρώσουν τον κύκλο ζωής τους. Οι mobile agents είναι μικροί, φτηνοί και επιβιώνουν για συγκεκριμένο, σύντομο χρονικό διάστημα. Όταν η τιμή του παράγοντα επικινδυνότητας πέσει σε επιτρεπτά όρια (κάτω από 50) , τότε παύει να υπολογίζεται ως εισβολή και τα mobile ants επικεντρώνονται μόνο στα σημεία που υπάρχει πρόβλημα.

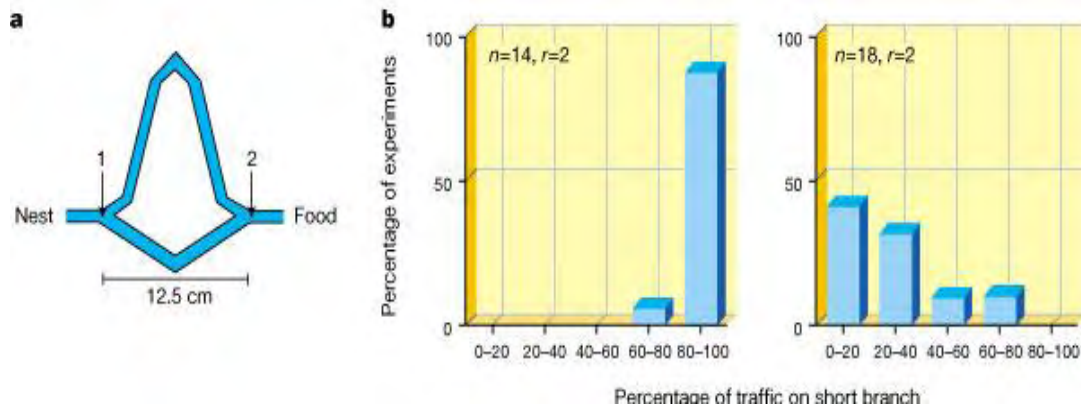
3.4.2 Προσαρμοστικότητα Network Simulator

Η προσαρμοστικότητα του Network Simulator είναι αντίστοιχη με αυτή του Ant Colony Simulator που αναπτύχθηκε στην ενότητα § 3.3.2

3.5 Σενάρια εκτέλεσης

3.5.1 Το πείραμα της διπλής γέφυρας

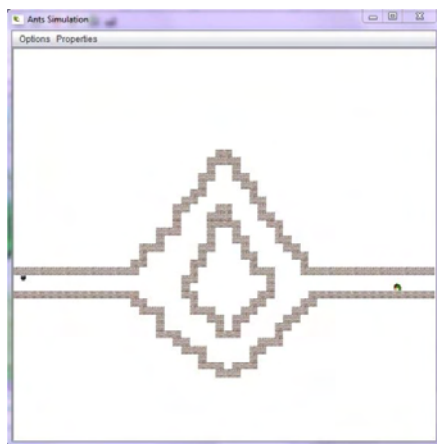
Ο Deneubourg και οι συνεργάτες του παρατήρησαν ότι αναζητώντας τροφή τα μυρμηγκία επιλέγουν τη συντομότερη διαδρομή μεταξύ φωλιάς και πηγής με τροφή. Την παρατήρηση αυτή μετέφεραν στα εργαστήριά τους, όπου εκτέλεσαν ένα πείραμα που το ονόμασαν “διπλή γέφυρα”. Σε αυτό υπήρχαν δύο εναλλακτικές διαδρομές που συνέδεαν την φωλιά των μυρμηγκιών με μια πηγή τροφής.



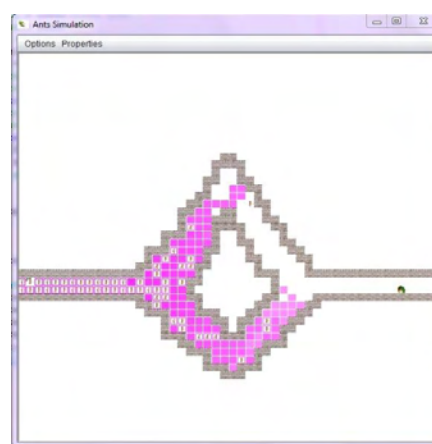
Εικόνα 3.24 Πείραμα διπλής γέφυρας

Εκτέλεσαν το πείραμα σε πολλές επαναλήψεις. Παρατήρησαν ότι το ποσοστό της επισκεψιμότητας του μικρότερου σε μήκος κλάδου (12,5 cm) ήταν πολύ μεγαλύτερο αυτού του δεύτερου μονοπατιού. Το αριστερό γράφημα (14 πειράματα) αντιπροσωπεύει πειράματα στα οποία τα δύο μονοπάτια ήταν διαθέσιμα στα μυρμηγκία ταυτόχρονα. Το δεξί γράφημα (18 πειράματα) αντιπροσωπεύει πειράματα στα οποία η μικρότερη διαδρομή (12,5 cm) έγινε διαθέσιμη στην αποικία 30 λεπτά μετά την μεγαλύτερη: επειδή η μικρότερη διαδρομή δεν έχει επιλεγεί και δεν έφερε ίχνη φερομόνης η αποικία δεν την επιλέγει τελικά.

Το συγκεκριμένο πείραμα δείχνει την σημαντικότητα της στιγμεργίας στις αποικίες των μυρμηγκιών. Γι'αυτό το λόγο αποτελεί κομμάτι της εφαρμογής μας. Στην κύρια εφαρμογή δημιουργούμε μία διπλή γέφυρα και τοποθετούμε δεξιά της μια ποσότητα τροφής και αριστερά την φωλιά παραγωγής μυρμηγκιών. Για $t=5$ βλέπουμε τα μυρμηγκία να κινούνται πιθανοτικά και να διαλέγουν άλλα το πάνω μονοπάτι και άλλα το κάτω αφήνοντας πίσω τους τη ροζ φερομόνη της φωλιάς.

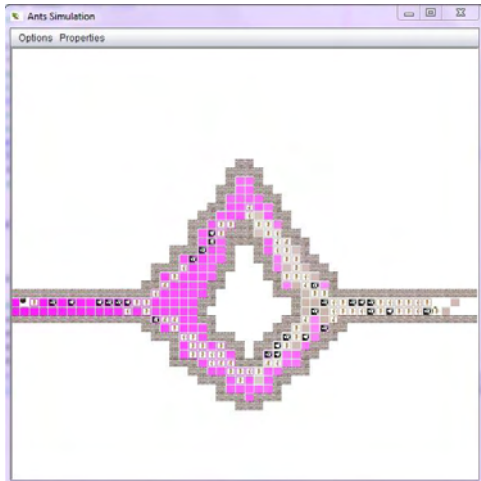


Εικόνα 3.25 Για $t=0$

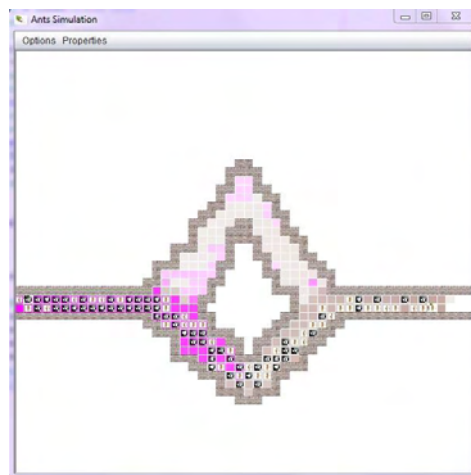


Εικόνα 3.26 Για $t=5$

Για $t=35$ βλέπουμε πως κάποια έχουν βρει την τροφή και γυρίζουν στην φωλιά είτε από το πάνω είτε από το κάτω μονοπάτι αφήνοντας γκρι φερομόνη. Ωστόσο με την πάροδο του χρόνου ($t = 50$) τελικά επιλέγεται το συντομότερο μονοπάτι που είναι το κάτω.

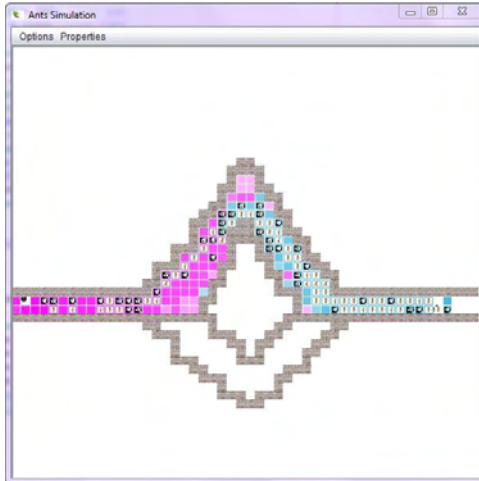


Εικόνα 3.27 Για $t=35$

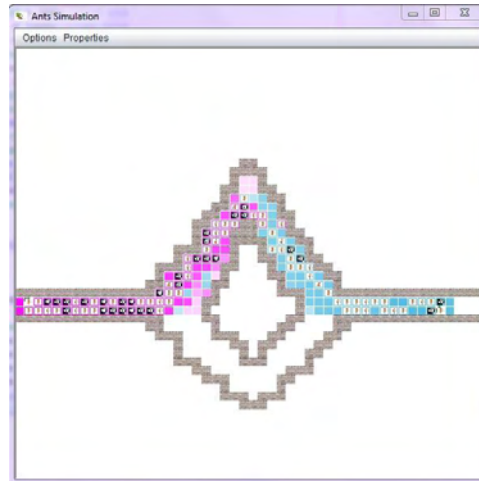


Εικόνα 3.28 Για $t=50$

Στην συνέχεια φράζουμε το κάτω μονοπάτι (συντομότερο) σε αναλογία του πειράματος Deneubourg και μετά από την πάροδο $t=50$ sec το κάνουμε διαθέσιμο.



Εικόνα 3.29 Για $t=30$

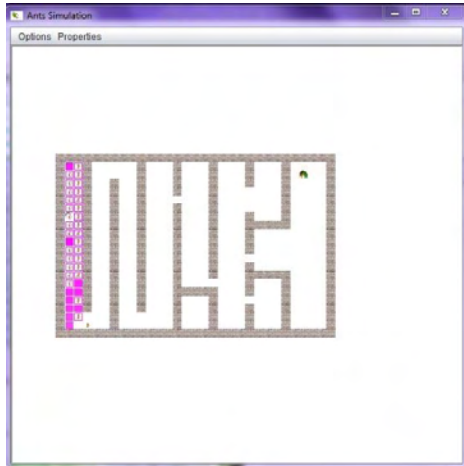


Εικόνα 3.30 Για $t=50$

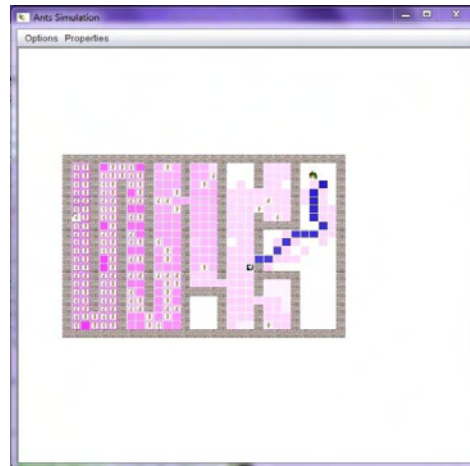
Όπως φαίνεται και στην εικόνα 3.30 παρόλο που το κάτω μονοπάτι είναι το συντομότερο επειδή δεν φέρει ίχνη φερομόνης τελικά δεν επιλέγεται. Καταλήξαμε δηλαδή σε όμοια συμπεράσματα με την ομάδα του Deneubourg .

3.5.2 Παράδειγμα λαβύρινθου

Επιπλέον λειτουργία του προσομοιωτή μας είναι η κατασκευή του παραδείγματος του λαβύρινθου. Στόχος μας είναι να δείξουμε πως ο ACO μπορεί να βρει το συντομότερο μονοπάτι ακόμα και σε ένα μη φιλικό περιβάλλον γεμάτο εμπόδια. Γι' αυτό κατασκευάσαμε έναν λαβύρινθο με στενούς διαδρόμους. Στο ένα άκρο τοποθετήσαμε τη φωλιά και στο άλλο την τροφή.

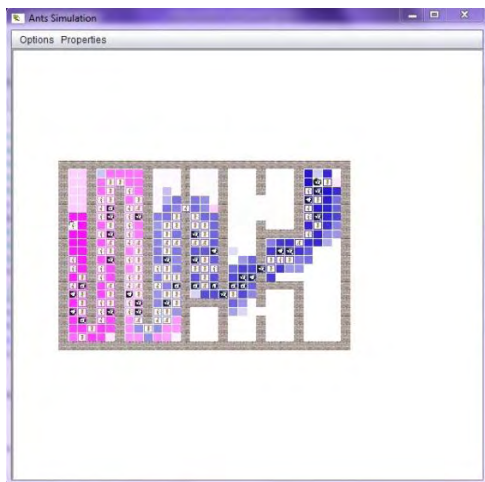


Εικόνα 3.31 Για $t=3$ sec



Εικόνα 3.32 Για $t=80$ sec

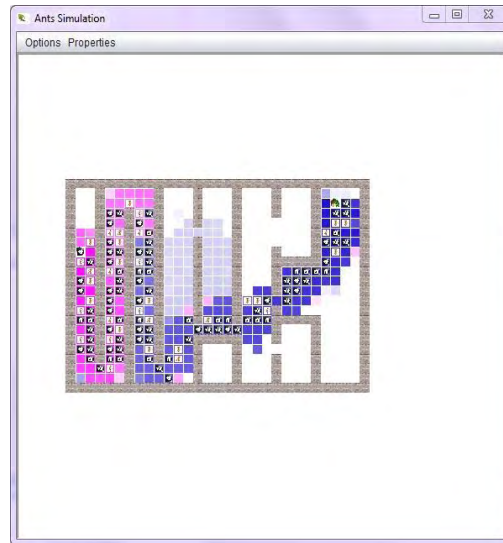
Βλέπουμε πως μετά από 80 sec ένα από τα μυρμήγκια φτάνει στην τροφή. Η εύρεση της τροφής ήταν πολύ δύσκολο έργο, αλλά τελικά ακολουθώντας τα ίχνη της φερομόνης βρίσκεται το μονοπάτι φωλιάς-τροφής. Αν στη συνέχεια ανοίξουμε έναν καινούριο συντομότερο δρόμο (αφαιρούμε το εμπόδιο που είναι κυκλωμένο με κόκκινο χρώμα εικόνα 3.34), που να είναι ωστόσο κοντά στο μονοπάτι φερομόνης, με το πέρασμα του χρόνου θα ακολουθηθεί το νέο και συντομότερο.



Εικόνα 3.33 Για $t=125$ sec



Εικόνα 3.34 Για $t=127$ sec



Εικόνα 3.35 Για $t = 132 \text{ sec}$

Η διαφορά με το παράδειγμα της διπλής γέφυρας είναι ότι στο πρώτο, όταν ο δεύτερος και πιο συντομότερος δρόμος έγινε διαθέσιμος κανένα από τα κελιά του δεν έφερε ίχνος φερομόνης για να ακολουθηθεί. Στο παράδειγμα του λαβυρίνθου όμως, επειδή τόσο δεξιά όσο και αριστερά του εμποδίου που επιλέγουμε να αφαιρέσουμε υπάρχει φερομόνη, τότε είναι πολύ πιθανόν κάποιο από τα μυρμήγκια που κινείτε πιθανοτικά να βρει την νέα αυτή εσοχή. Έτσι όλο και περισσότερα μυρμήγκια θα ακολουθούν το νέο μονοπάτι, ενώ σταδιακά η φερομόνη από το παλιό (Εικόνα 3.35) θα εξατμιστεί².

3.6 Πειραματισμός με ιδιότητες του συστήματος

Σημαντικές μεταβλητές που επηρεάζουν το σύστημά μας είναι οι :

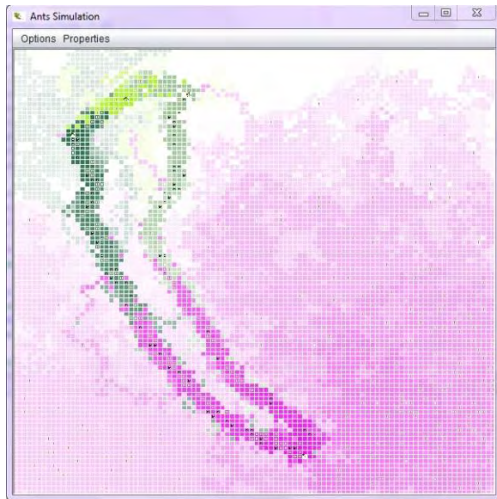
1. **Pheromone** : η τιμή της φερομόνης που αφήνει κάθε μυρμήγκι στο πέρασμα του αφού βρει μια πηγή τροφής (τιμές 10-100)
2. **Evaporation**: η μεταβλητή αυτή ανάλογα με την αυξομείωσή της οδηγεί στην πιο γρήγορη ή πιο αργή εξασθένιση της φερομόνης. Μεγάλες τιμές της evaporation κάνουν την φερομόνη να εξασθενεί σχεδόν αμέσως (τιμές 1-50)
3. **DecreaseNP**: η μεταβλητή αυτή αφορά τον ρυθμό εξάτμισης της ροζ φερομόνης, αυτής που αφήνουν τα μυρμήγκια για να βρουν τον δρόμο του γυρισμού στη φωλιά (τιμές 5 - 40)
4. **Follow Strict**: η μεταβλητή αυτή δείχνει πόσο αυστηρά ακολουθούν τα μυρμήγκια ένα μονοπάτι ή κινούνται πιο ελεύθερα. Όσο μεγαλύτερη είναι η τιμή τόσο πιο συγκεκριμένο γίνεται ένα μονοπάτι χωρίς παρεκκλίσεις (μειώνεται η πιθανότητα της τυχαίας επιλογής επόμενου κελιού)

² Τα στιγμιότυπα μπορούν να μεταβληθούν μιας και το μοντέλο μας είναι πιθανοτικό γι' αυτό και οι χρονικές στιγμές είναι ενδεικτικές χωρίς να έχουμε ωστόσο μεγάλες αποκλίσεις σε διαφορετικές εκτελέσεις

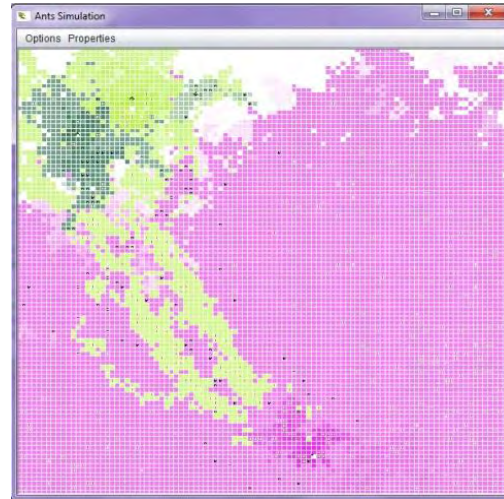
Για την καλύτερη κατανόηση των μεταβλητών θα δείξουμε κάποια στιγμιότυπα για κάθε μεταβλητή.

3.6.1 Pheromone

Evaporation = 4, DecreaseNP = 5 και Follow Strict =70

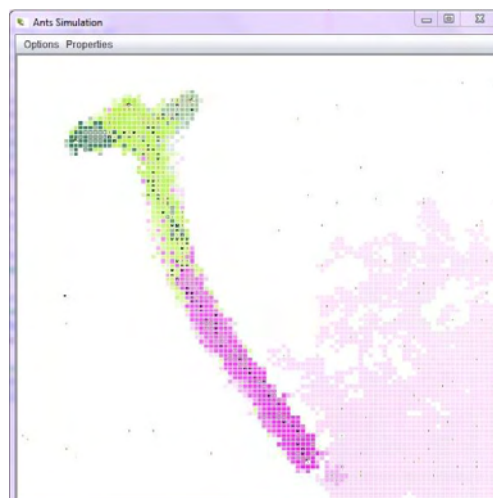


Εικόνα 3.36 Pheromone = 100



Εικόνα 3.37 Pheromone = 10

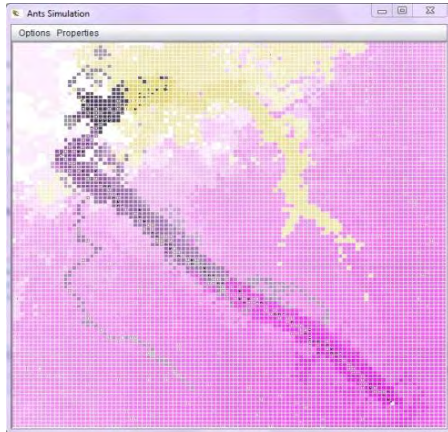
Αλλάζοντας την τιμή της φερομόνης από 100 σε 10 και κρατώντας τις υπόλοιπες μεταβλητές ίδιες, βλέπουμε πως οι φερομόνες που αφήνουν τα μυρμήγκια μετά την εύρεση της τροφής εξασθενούν και τελικά εξαφανίζονται. Έτσι ουσιαστικά χάνεται το μονοπάτι μεταξύ τροφών και φωλιάς και τα μυρμήγκια είναι αναγκασμένα να βρουν νέο μονοπάτι γεγονός που πραγματοποιείται αρκετές χρονικές στιγμές αργότερα.



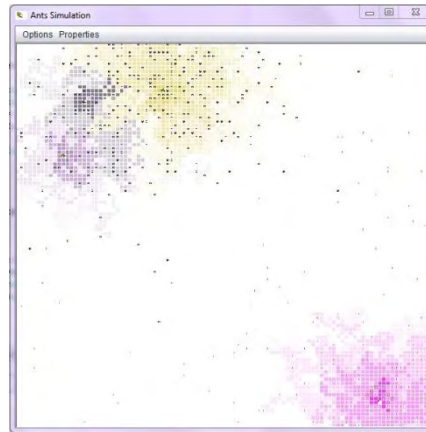
Εικόνα 3.38 t=260 sec

3.6.2 Evaporation

Pheromone = 100, DecreaseNP = 5 και Follow Strict =70



Εικόνα 3.39 Evaporation = 1

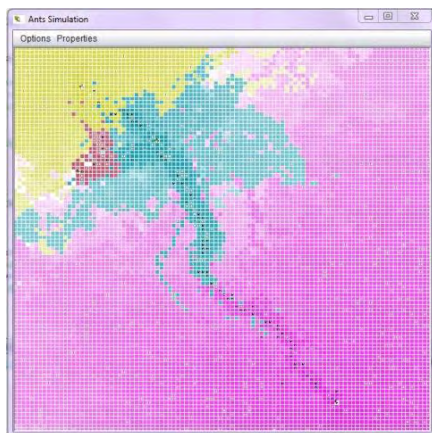


Εικόνα 3.40 Evaporation = 50

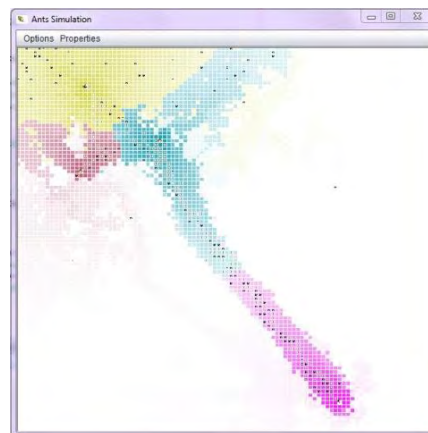
Αυξάνοντας την μεταβλητή που επηρεάζει τον ρυθμό με τον οποίο εξασθενεί η φερομόνη τόσο των μυρμηγκιών που φεύγουν από την φωλιά όσο και αυτών που έχουν βρει τροφή, παρατηρούμε πως ουσιαστικά χάνεται το μονοπάτι που πριν ένωνε τις τροφές με την φωλιά. Αυτό με την πάροδο του χρόνου δεν αλλάζει. Τα μυρμήγκια δεν βρίσκουν ποτέ δρόμο πίσω προς την φωλιά. Ουσιαστικά ο αλγόριθμος δεν χρησιμοποιείται και όλα τα μυρμήγκια κινούνται εντελώς πιθανοτικά χωρίς να λαμβάνουν υπόψη τους τον αλγόριθμο ACO στην απόφαση, για το επόμενο κελί που θα ακολουθήσουν.

3.6.3 DecreaseNP

Pheromone = 100, Evaporation = 1 και Follow Strict =70



Εικόνα 3.41 DecreaseNP = 5

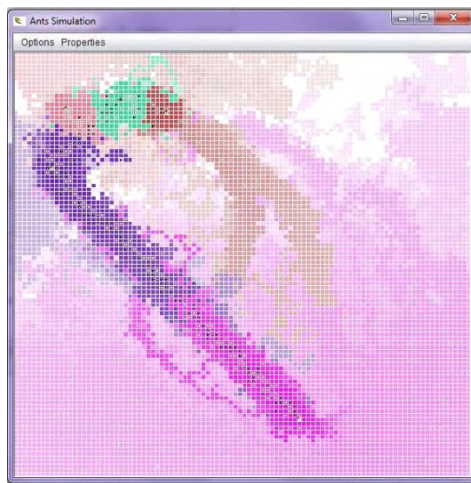


Εικόνα 3.42 DecreaseNP =40

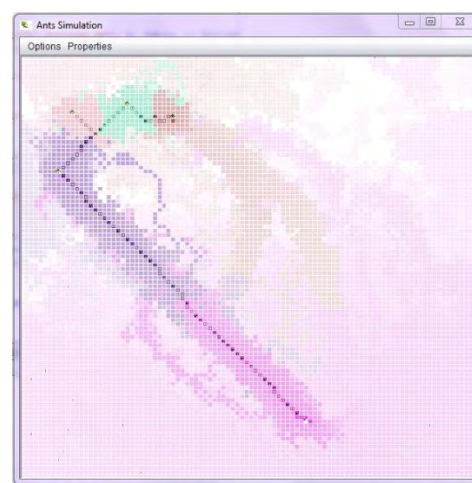
Όταν η μείωση της έντασης της ροζ φερομόνης είναι μικρή βλέπουμε πόσο έντονη είναι η διάχυση της σε όλο το πλέγμα. Όταν όμως γίνει πολύ μεγάλη τότε η ροζ φερομόνη σχεδόν εξαφανίζεται και έτσι αρκετά από τα μυρμηγκία δεν βρίσκουν μονοπάτι για να γυρίσουν. Ωστόσο η φερομόνη δεν εξαφανίζεται σε τέτοιο βαθμό που να χάνεται εντελώς το μονοπάτι φωλιάς-τροφών.

3.6.3 Follow Strict

Pheromone = 100, Evaporation = 1 και DecreaseNP = 5



Εικόνα 3.42 Follow Strict =50



Εικόνα 3.43 Follow Strict =100

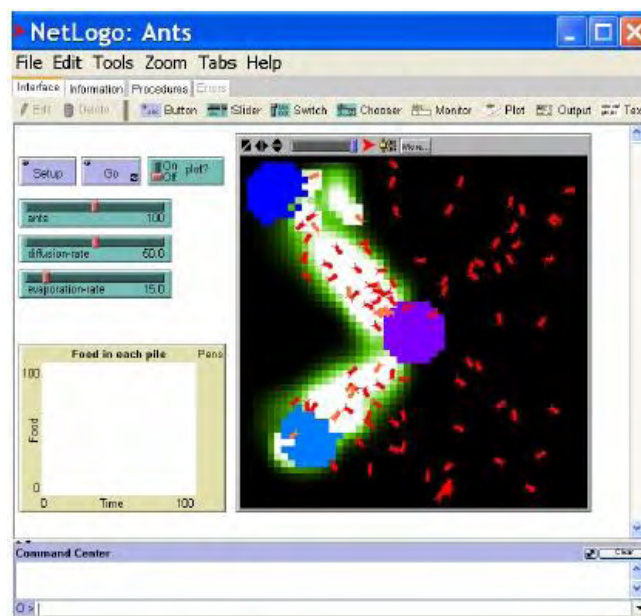
Κάνοντας πιο αυστηρό το σύστημά μας καταλήγουμε σε ένα πιο συγκεκριμένο μονοπάτι. Τα μυρμηγκία ακολουθούν πιστά το συγκεκριμένο πρώτο βέλτιστο μονοπάτι. Λόγω της έλλειψης της πιθανοτικής συμπεριφοράς των μυρμηγκιών έχουμε σαν αποτέλεσμα τα μυρμηγκία να μην ακολουθήσουν ένα μελλοντικά συντομότερο βέλτιστο μονοπάτι γιατί απλά, δεν θα έχουν την πιθανότητα να το ανακαλύψουν.

3.7 Παρόμοιες εφαρμογές προσομοίωσης

Υπάρχουν διάφοροι simulators της συμπεριφοράς που αναπτύσσουν οι αποικίες των μυρμηγκιών. Όλοι παρουσιάζουν παρόμοιες λειτουργίες.

3.7.1 NetLogo

NetLogo είναι ένα απλό περιβάλλον προσομοίωσης που βασίζεται στην γλώσσα προγραμματισμού StarLogo. Στο NetLogo, το περιβάλλον είναι δυναμικό και μπορεί να υποστηρίξει συστήματα που βασίζονται στην στιγμεργία. Επιπλέον, μπορεί εύκολα να τροποποιηθεί και οι πράκτορες μπορούν να λάβουν πληροφορίες για τις γειτονικές περιοχές τους, χρησιμοποιώντας διάφορους αισθητήρες. Ο προγραμματιστής δεν έχει έλεγχο των agents, αυτοί λειτουργούν και κινούνται αυτόνομα.

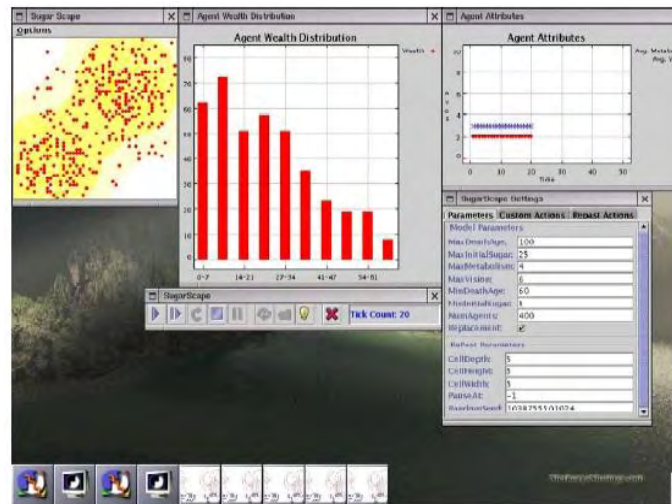


Εικόνα 3.44 NetLogo Simulator

Περισσότερες πληροφορίες καθώς και δυνατότητα εγκατάστασης υπάρχουν στην διεύθυνση <http://ccl.northwestern.edu/netlogo/>.

3.7.2 Repast

Το Recursive Porous Agent Simulation Toolkit (Repast) είναι ένα πρόγραμμα βασισμένο στην γλώσσα προγραμματισμού Java. Είναι ένα ευρέως χρησιμοποιούμενο δωρεάν, cross-platform εργαλείο προσομοίωσης. Μπορεί να χρησιμοποιηθεί από χρήστες που προέρχονται από διαφορετικές χώρες, μιας και προσφέρεται σε πολλές γλώσσες. Ενσωματώνει λειτουργίες που παρουσιάζουν κυρίως οι γενετικοί αλγόριθμοι

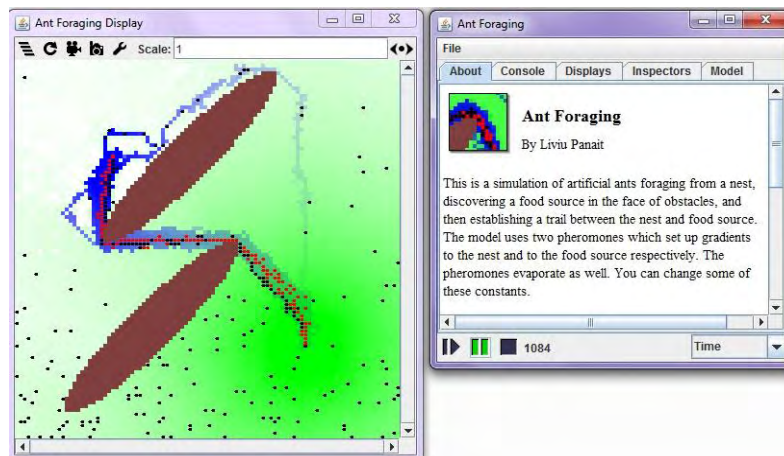


Εικόνα 3.45 Repast Simulator

Περισσότερες πληροφορίες καθώς και δυνατότητα εγκατάστασης υπάρχουν στην διεύθυνση <http://repast.sourceforge.net/>:

3.7.3 Mason

Η εφαρμογή MANSON (Multi-Agent Simulator Of Neighborhoods/Networks) είναι αποτέλεσμα της συνεργασίας του πανεπιστημίου George Mason με το κέντρο κοινωνικής πολυπλοκότητας GMU. Η παρατήρηση του τρόπου αναζήτησης της τροφής από τα μυρμηγκία είναι απλά ένα κομμάτι της εφαρμογής. Το περιβάλλον δράσης των μυρμηγκιών είναι στατικό και ο χρήστης είναι δύσκολο να προσθέτει τροφές ή εμπόδια.



Εικόνα 3.46 Mason Simulator

Περισσότερες πληροφορίες καθώς και δυνατότητα εγκατάστασης υπάρχουν στην διεύθυνση <http://cs.gmu.edu/~eclab/projects/mason/>

3.8 Μειονεκτήματα και μελλοντική εξέλιξη της εφαρμογής

Στο σημείο αυτό, είναι σημαντικό να αναφέρουμε κάποια μειονεκτήματα που παρουσιάζει η εφαρμογή μας. Δημιουργούμε ένα πλέγμα 50x50 ή 100x100 μέσα στο οποίο υπάρχουν κελιά. Τα κελιά αυτά ανανεώνονται συνέχεια επειδή έχουμε ένα δυναμικό περιβάλλον που μεταβάλλεται συνεχώς. Για να προσομοιώσουμε τις οντότητες ANT-μυρμήγκι, OBSTACLE-εμπόδιο, FOOD-τροφή, DANGER-κίνδυνος, WIRELESS LAN-ασύρματο δίκτυο και σταθερό δίκτυο-LAN χρησιμοποιήσαμε εικόνες. Αυτό έχει σαν αποτέλεσμα όταν εκτελούμε την εφαρμογή μας αυτή σταδιακά επιβραδύνεται αφού και χρειάζεται όλο και περισσότερο χρόνο για να ανανεωθεί το πλέγμα κελιών που περιέχον εικόνες. Ένα άλλο μειονέκτημα είναι πως η εφαρμογή δεν είναι εντελώς δυναμική. Δηλαδή ο χρήστης χειροκίνητα εισάγει κάποια τροφή, κάποιον κίνδυνο ή κάποιο εμπόδιο.

Μια από τις μελλοντικές εξελίξεις της εφαρμογής είναι ο περιορισμός του χρήστη. Οι τροφές, κίνδυνοι και εμπόδια θα μπορούσαν να εμφανίζονται δυναμικά ανά τακτά χρονικά διαστήματα. Στόχος για το μέλλον αποτελεί και η εισαγωγή κατάλληλης μεταβλητής χρόνου που θα υπολογίζει τον χρόνο που έκαναν τα μυρμήγκια για να βρουν το βέλτιστο μονοπάτι. Ιδανικός στόχος, τέλος, είναι η υλοποίηση και χρήση των παραλλαγών/εξελίξεων του ACO αλγορίθμου που παρουσιάστηκαν στην ενότητα § 2.7.2 ώστε να μετατραπεί η εφαρμογή μας σε ένα εργαλείο παρατήρησης των Ant-based αλγορίθμων.

Κεφάλαιο 4ο

4 Αλγόριθμος ACO σε ad-hoc δίκτυα

4.1 Εισαγωγή

Τα τελευταία χρόνια παρατηρείται η συνεχώς αυξανόμενη χρήση αδόμητων ασύρματων (ad-hoc) δικτύων για την υποστήριξη διαφορετικών εφαρμογών όπως είναι στον τομέα της επικοινωνίας, της υγείας, της εκπαίδευσης, του περιβάλλοντος, της κυκλοφοριακής κίνησης και αλλού. Σε ένα τέτοιο περιβάλλον, αναμένεται ότι η υπολογιστική νοημοσύνη θα εμποδωθεί στον περιβάλλοντα χώρο των χρηστών αντί σε κάποιες εξειδικευμένες ειδικές συσκευές. Η αυτοοργάνωση κινητών κόμβων σε δίκτυα ευρείας κλίμακας, τους ενισχύει με την ικανότητα να συνεργάζονται σε μεγαλύτερες και πολυπλοκότερες διεργασίες, και αναμένεται να προσδώσει μια επαναστατική διάσταση στις διαδικασίες επικοινωνίας, συλλογής και επεξεργασίας δεδομένων. Συνεπώς κάτω από αυτές τις εξελίξεις και τάσεις που διαμορφώνονται, η μελέτη των ad-hoc δικτύων λαμβάνοντας υπόψη τα βασικά και διακριτά χαρακτηριστικά της συμπεριφοράς αυτών, καθίσταται όχι μόνο επιθυμητή αλλά απόλυτα επίκαιρη και αναγκαία.

Τα αδόμητα ασύρματα (ad-hoc) δίκτυα αποτελούνται από κόμβους που συνδέονται δυναμικά μεταξύ τους και μέσω ασύρματων συνδέσμων. Χαρακτηριστικό των ad-hoc δικτύων είναι ότι δεν υποθέτουν την ύπαρξη κάποιας σταθερής υποδομής, σε αντίθεση με τα υπόλοιπα δίκτυα που απαιτούν αυτήν την υποδομή με τη μορφή των σταθμών βάσης. Οι κόμβοι ενός ad-hoc δικτύου μπορεί να είναι κινητοί ή σταθεροί και να έχουν σταθερή ή μεταβαλλόμενη ακτίνα μετάδοσης.

Το κύριο χαρακτηριστικό των ad-hoc δικτύων αλλά και ταυτόχρονα η διαφορά από τα υπόλοιπα ασύρματα δίκτυα, είναι πως δεν στηρίζονται σε υπάρχουσα δικτυακή υποδομή. Σε θέματα δρομολόγησης και διαχείρισης παρατηρείται μια αυτοοργάνωση και αυτοδιαχείριση του δικτύου από τους ίδιους τους κόμβους του. σε σχέση με τα συνηθισμένα ασύρματα δίκτυα. Και αυτό γιατί αυτοί οι κόμβοι σε ένα δίκτυο τέτοιου είδους μπορούν να λειτουργήσουν και σαν δρομολογητές (routers) αλλά και σαν εξυπηρετητές (servers), να προωθούν πακέτα εκ μέρους άλλων κόμβων και να τρέχουν εφαρμογές χρηστών.

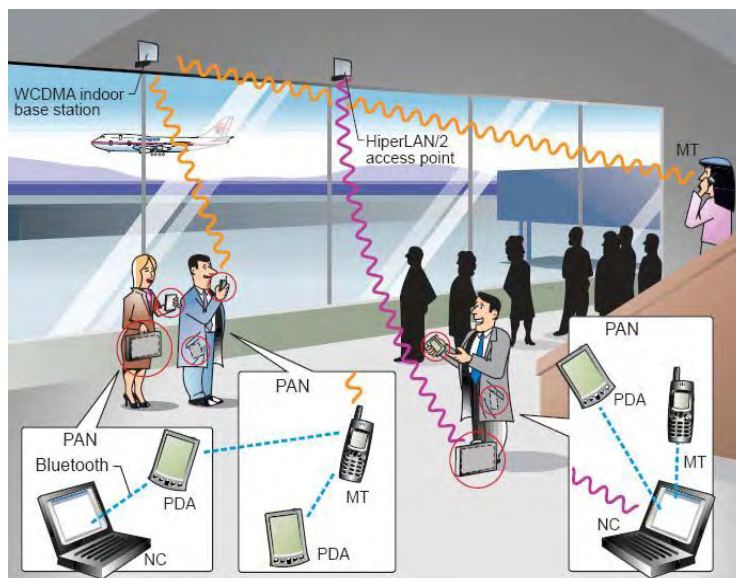
4.2 Κινητά ad hoc δίκτυα (MANET)

Η γρήγορη εγκατάσταση και η ελάχιστη απαιτούμενη διαμόρφωση καθιστά τα ασύρματα ad hoc δίκτυα κατάλληλα για καταστάσεις έκτακτης ανάγκης, όπως φυσικές καταστροφές ή πολεμικές συρράξεις. Η παρουσία δυναμικών και προσαρμοστικών πρωτοκόλλων επιτρέπει στα ad hoc δίκτυα να σχηματίζονται γρήγορα.

Τα ασύρματα ad hoc δίκτυα μπορούν να ταξινομηθούν περαιτέρω με βάση την εφαρμογή τους:

- Κινητά ad hoc δίκτυα (MANET - Mobile Ad hoc NETWORKs)
- Ασύρματα δίκτυα πλέγματος (WMN - Wireless Mesh Networks)
- Ασύρματα δίκτυα αισθητήρων (WSN - Wireless Sensor Networks)

Τα δίκτυα Mobile Ad-Hoc Networks (MANET) λοιπόν, είναι μια επέκταση των δικτύων Ad-Hoc, αφού αποτελούνται από κόμβους που κινούνται δυναμικά και αυθαίρετα, δημιουργώντας έτσι δυναμικές τοπολογίες δικτύου



Εικόνα 4.1 Δίκτυο MANET

Σήμερα, η ιδέα μας για τα Mobile Ad-Hoc δίκτυα περιλαμβάνει σενάρια όπως αυτά που απεικονίζονται στην Εικόνα 4.1, όπου άνθρωποι έχουν συσκευές που μπορούν να επικοινωνούν με Ad-Hoc δίκτυα. Οι συσκευές ενός χρήστη μπορούν να διασυνδεθούν μεταξύ τους και να συνδέονται σε τοπικά κέντρα πληροφόρησης. Για παράδειγμα, για να αντλούν ενημερώσεις για τις αναχωρήσεις των πτήσεων, αλλαγές στις πύλες εξόδου και άλλα. Οι Ad-Hoc συσκευές μπορούν επίσης να προωθήσουν δικτυακή κίνηση σε συσκευές που είναι εκτός εμβέλειας

4.2.1 Χαρακτηριστικά των δικτύων MANET

Ένα δίκτυο MANET αποτελείται από κινητές μονάδες (π.χ. ένα δρομολογητή με πολλούς hosts και ασύρματες συσκευές, που αποκαλούνται κόμβοι) οι οποίες είναι ελεύθερες να μετακινηθούν όπου θέλουν. Αυτοί οι κόμβοι μπορούν να βρίσκονται σε αεροπλάνα, πλοία, φορτηγά, αυτοκίνητα, ακόμα και σε ανθρώπους. Ένα δίκτυο MANET λοιπόν, είναι ένα αυτόνομο σύστημα αποτελούμενο από κινητούς κόμβους. Το σύστημα αυτό μπορεί να λειτουργεί απομονωμένο, ή να έχει και διεξόδους (gateways) και να επικοινωνεί με ένα σταθερό δίκτυο. Στον δεύτερο τρόπο λειτουργίας, το σύστημα θα λειτουργεί σαν ένα «αποκομμένο δίκτυο» (“stub” network) που συνδέεται με ένα σταθερό δίκτυο. Τα «αποκομμένα δίκτυα» μεταφέρουν δικτυακή κίνηση που προέρχεται ή κατευθύνεται προς τους εσωτερικούς κόμβους, αλλά δεν επιτρέπει εξωτερική κίνηση να μεταφερθεί μέσω του «αποκομμένου δικτύου».

Οι κόμβοι του δικτύου MANET είναι εξοπλισμένοι με ασύρματους πομπούς και δέκτες χρησιμοποιώντας κεραίες που μπορεί να είναι μη κατευθυντήριες (omnidirectional), πολύ κατευθυντήριες (point-to-point), πιθανώς μεταβλητές, ή κάποιος συνδυασμός των παραπάνω. Σε κάποιο χρονικό σημείο, ανάλογα με τη θέση των κόμβων, την εμβέλεια των πομποδεκτών τους, τη μεταδιδόμενη ισχύ τους και τα επίπεδα παρεμβολών, μια ασύρματη σύνδεση στη μορφή ενός τυχαίου “Ad-hoc” δικτύου δημιουργείται ανάμεσά τους. Αυτή η Ad-hoc τοπολογία μπορεί να αλλάξει με την πάροδο του χρόνου, καθώς οι κόμβοι μετακινούνται ή αλλάζουν την ισχύ μετάδοσής τους.

Τα δίκτυα MANET έχουν πολλά αξιοπρόσεκτα χαρακτηριστικά :

1. **Δυναμικές τοπολογίες.** Οι κόμβοι είναι ελεύθεροι να μετακινούνται αυθαίρετα, με αποτέλεσμα η τοπολογία του δικτύου, που είναι συνήθως multihop, να μπορεί να αλλάζει τυχαία και γρήγορα σε απρόβλεπτες στιγμές. Επίσης η τοπολογία μπορεί να απαρτίζεται και από αμφίδρομες (bidirectional) αλλά και από μονοκατευθυντικές (unidirectional) ζεύξεις.
2. **Ζεύξεις περιορισμένου εύρους ζώνης και μεταβλητής χωρητικότητας.** Οι ασύρματες ζεύξεις θα συνεχίσουν να έχουν σημαντικά μικρότερες χωρητικότητες από τις αντίστοιχες ενσύρματες. Επιπρόσθετα, το πραγματικό throughput στις ασύρματες επικοινωνίες, αν λάβουμε υπόψη τα φαινόμενα της πολλαπλής πρόσβασης (multiple access), του θορύβου, των παρεμβολών, κτλ. είναι συνήθως πολύ μικρότερο από τη μέγιστη εκπομπή των πομπών. Ένα αποτέλεσμα της σχετικά χαμηλής έως μέτριας χωρητικότητας των ζεύξεων είναι ο συνωστισμός, που τυπικά είναι ο κανόνας και όχι η εξαίρεση. Π.χ. η αυξανόμενη ζήτηση μιας δικτυακής εφαρμογής πιθανότατα θα πλησιάσει ή και θα ξεπεράσει τη χωρητικότητα του δικτύου. Αφού το κινητό δίκτυο είναι συχνά μια επέκταση ενός σταθερού δικτύου, οι χρήστες του κινητού δικτύου απαιτούν παρόμοιες υπηρεσίες με αυτές του σταθερού δικτύου. Οι απαιτήσεις

αυτές θα συνεχίσουν να αυξάνονται με την ανάπτυξη των πολυμέσων και των δικτυακών εφαρμογών.

3. **Λειτουργία με περιορισμένη ενέργεια τροφοδοσίας.** Μερικοί ή όλοι οι κόμβοι σε ένα δίκτυο MANET μπορεί να βασίζονται σε μπαταρίες ή σε άλλα εξαντλήσιμα μέσα τροφοδοσίας. Για αυτούς τους κόμβους, το πιο σημαντικό κριτήριο σχεδιασμού για βελτίωση είναι η κατανάλωση ενέργειας.
4. **Περιορισμένη ασφάλεια του φυσικού μέσου μετάδοσης (physical layer).** Τα κινητά ασύρματα δίκτυα είναι γενικά πιο ευπαθή σε θέματα ασφάλειας του φυσικού μέσου μετάδοσης σε σχέση με τα ασύρματα δίκτυα. Οι αυξημένες πιθανότητες για επιθέσεις τύπου eavesdropping, spoofing και DoS (Denial-of-Service) θα πρέπει να ληφθούν σοβαρά υπόψη. Συχνά εφαρμόζονται ήδη υπάρχοντες τεχνικές ασφάλειας στα ασύρματα δίκτυα για την μείωση των τρωτών σημείων στην ασφάλεια. Η αποκεντρωμένη φύση της διαχείρισης των δικτύων MANET παρέχει επιπλέον κάλυψη για περιπτώσεις που κάποιος κόμβος βγει εκτός λειτουργίας (single point failure) σε σχέση με τις πιο συγκεντρωμένες (centralized) προσεγγίσεις.
5. **Κλιμάκωση (Scalability).** Σε κάποια πιθανά δίκτυα MANET όπως στρατιωτικά δίκτυα ή δίκτυα σε αυτοκινητόδρομους, ο αριθμός των κόμβων ενδέχεται να είναι σχετικά μεγάλος, μερικές δεκάδες ή ακόμα και εκατοντάδες κόμβοι ανά περιοχή δρομολόγησης (routing area), επομένως απαιτείται η υποστήριξη κλιμάκωσης σε αυτά τα δίκτυα. Μπορεί η ανάγκη για κλιμάκωση να μην είναι μοναδική για τα MANET, αλλά οι μηχανισμοί για την επίτευξή της είναι.

Αυτά τα χαρακτηριστικά δημιουργούν ένα σύνολο προϋποθέσεων για τον σχεδιασμό των πρωτοκόλλων που επεκτείνονται πολύ πιο γρήγορα σε σχέση με τα πρωτόκολλα σχεδόν σταθερής τοπολογίας του Internet.

4.2.2 Εφαρμογές των δικτύων MANET

Τα MANET έχουν πρακτική εφαρμογή σε περιπτώσεις όπου δεν υπάρχει κάποια σταθερή ενσύρματη δικτυακή υποδομή (fixed wired infrastructure). Τέτοιες περιπτώσεις έχουμε όταν δεν είναι οικονομικά, πρακτικά ή γεωγραφικά εφικτό να δημιουργηθεί η απαραίτητη υποδομή, ή επειδή οι καταστάσεις δεν επιτρέπουν την εγκατάστασή της, όπως :

- Σε μια συνεδριακή αίθουσα κατά τη διάρκεια συναντήσεων, όταν οι συμμετέχοντες θέλουν να ανταλλάξουν πληροφορίες.
- Σε μια αίθουσα διδασκαλίας κατά τη διάρκεια συζητήσεων με τον καθηγητή, ή και κατά τη διάρκεια της διδασκαλίας.
- Σε ένα διαφημιστικό περιβάλλον
- Σε ένα αεροδρόμιο όπου οι εργαζόμενοι θέλουν να ανταλλάξουν αρχεία.
- Σε μια επείγουσα επιχείρηση διάσωσης, όταν τα μέλη του σωστικού συνεργείου θέλουν να συντονίσουν την προσπάθειά τους. Για παράδειγμα σε

περίπτωση όταν κάποιος σεισμός ή πλημμύρα καταστρέψει την ενσύρματη υποδομή των σταθερών δικτύων.

- Σε μάχες κατά τη διάρκεια πολέμου, για τον συντονισμό των στρατιωτών στην άμυνα και την επίθεση.

4.3 Πρωτόκολλα δρομολόγησης σε δίκτυα MANET

Ένα πρωτόκολλο δρομολόγησης (routing protocols) ορίζεται ως ένα σύνολο κανόνων που είναι υπεύθυνοι για ανακάλυψη και καθιέρωση των πιο αξιόπιστων και σύντομων μονοπατιών μεταξύ των κόμβων. Ταυτόχρονα, ένα πρωτόκολλο δρομολόγησης είναι υπεύθυνο για διατήρηση και επισκευή (όπου χρειάζεται) οποιουδήποτε μονοπατιού. Τα πρωτόκολλα δρομολόγησης χωρίζονται σε πέντε κύριες κατηγορίες:

- δυναμικά (proactive)
- αντιδραστικά (reactive)
- υβριδικά (hybrid)
- ιεραρχικά (hierarchical) και
- γεωγραφικά (geographical)

4.3.1 Δυναμικά πρωτόκολλα δρομολόγησης

Σε ένα δυναμικό πρωτόκολλο δρομολόγησης όλες οι διαδρομές προς κάθε προορισμό διατηρούνται σε έναν ενημερωμένο (up-to-date) πίνακα που διαχειρίζεται την κίνηση και την ορθότητα στις κατευθύνσεις των μονοπατιών. Έτσι οι αλλαγές στην τοπολογία του δικτύου ανανεώνονται συνεχώς καθώς εμφανίζονται. Εφαρμόζεται σε τοπολογίες που παρέχουν εναλλακτικές διαδρομές και έχει σαν κύριο γνώμονα την απόφαση που πρέπει να ληφθεί σχετικά με το ποια διαδρομή θα ακολουθήσουν τα προς αποστολή δεδομένα

4.3.2 Αντιδραστικά πρωτόκολλα δρομολόγησης

Στο αντιδραστικό πρωτόκολλο δρομολόγησης μια σύνδεση μεταξύ δύο κόμβων δημιουργείται μόνο όταν ζητηθεί από μια πηγή. Όταν μια διαδρομή εντοπίζεται, διατηρείται με μια διαδικασία συντήρησης έως ότου ο προορισμός θα υφίσταται. Τα αντιδραστικά πρωτόκολλα χρησιμοποιούν κάποιο αντιδραστικό αλγόριθμο και βρίσκουν μια διαδρομή κατόπιν απαίτησης, «πλημμυρίζοντας» το δίκτυο με πακέτα αιτημάτων για διαδρομές. Ωστόσο, η τεχνική αυτή έχει κάποια μειονεκτήματα σε θέματα καθυστέρησης στην ανακάλυψη διαδρομών και απόφραξης του δικτύου από τη συμφόρηση που προκαλείται λόγω της «πλημμύρας». Η πλημμύρα σε ένα δίκτυο αναφέρεται στο φαινόμενο όπου ένας δρομολογητής στέλνει ένα πακέτο από οποιοδήποτε κόμβο σε κάθε άλλο κόμβο του

δικτύου που είναι συνδεδεμένος με αυτόν, με σκοπό τη διατήρηση των ανανεώσιμων πληροφοριών σε μεγάλα δίκτυα.

4.3.3 Υβριδικά πρωτόκολλα δρομολόγησης

Τα υβριδικά πρωτόκολλα προσπαθούν να συνδυάσουν τα πλεονεκτήματα που εμφανίζουν τα αντιδραστικά και τα δυναμικά πρωτόκολλα, χρησιμοποιώντας τοπικά προδραστική δρομολόγηση και διατοπικά διαδραστική δρομολόγηση. Η επιλογή αυτή οφείλεται κυρίως στην υπόθεση αν ένας κόμβος εισέλθει στο δίκτυο ή το εγκαταλείψει, επηρεάζει μόνο την γειτονία του και όχι όλο το δίκτυο. Τέτοια πρωτόκολλα είναι μεταξύ άλλων το AntHocNet που στηρίζεται στον ACO αλγόριθμο και παρουσιάζεται παρακάτω.

4.3.4 Ιεραρχικά πρωτόκολλα δρομολόγησης

Η Ιεραρχική δρομολόγηση αναφέρεται σε ετερογενή δίκτυα επικοινωνιών, δηλαδή σε δίκτυα με ασύρματους και ενσύρματους κόμβους. Τα πλεονεκτήματα της ιεραρχικής δρομολόγησης εντοπίζονται ως προς την επεκτασιμότητα και την οικονομική επικοινωνία. Για την ενεργειακά οικονομική δρομολόγηση, τα πρωτόκολλα αυτού του είδους χρησιμοποιούνται στα «ασύρματα δίκτυα αισθητήρων». Οι κόμβοι υψηλής ενέργειας χρησιμοποιούνται για την επεξεργασία και την προώθηση των δεδομένων, ενώ αντίθετα οι κόμβοι χαμηλής ενέργειας χρησιμοποιούνται για μετρήσεις. Έχει αποδειχθεί ότι η ιεραρχική δρομολόγηση συμβάλλει ουσιαστικά όχι μόνο στην επεκτασιμότητα του δικτύου επικοινωνίας, αλλά και στην επεκτασιμότητα της διάρκειας ζωής του.

Τα πρωτόκολλα ιεραρχικής δρομολόγησης τα οποία παρουσιάζουν κάποιο ιδιαίτερο ενδιαφέρον ποικίλουν. Ωστόσο, κάποια από αυτά είναι το πρωτόκολλο “*Small Minimum Energy Communication Network*”, το οποίο είναι σε θέση να υπολογίσει ένα ενεργειακά οικονομικό υποδίκτυο, το οποίο εγγυάται τα μονοπάτια ελάχιστου κόστους και τις οικονομικές μεταδόσεις. Ενώ, το πρωτόκολλο “*Self Organizing Protocol*” το οποίο έχει την ικανότητα της αυτό-οργάνωσης, μπορεί να χρησιμοποιηθεί σε ετερογενή δίκτυα.

4.3.5 Γεωγραφικά πρωτόκολλα δρομολόγησης

Στην γεωγραφική δρομολόγηση οι αλγόριθμοι δρομολόγησης εξαλείφουν κάποια από τα προβλήματα των βασισμένων στην τοπολογία αλγορίθμων, χρησιμοποιώντας κάποιες επιπλέον πληροφορίες. Συγκεκριμένα, όπως δηλώνεται και από το όνομα τους, απαιτούν να γνωρίζουν την γεωγραφική (φυσική) θέση των κόμβων. Συνήθως, κάθε κόμβος ενημερώνεται για τη θέση του μέσω κάποιας υπηρεσίας εντοπισμού θέσης, όπως το GPS. Ο αποστολέας ενός πακέτου προσδιορίζει τη θέση του παραλήπτη με τη βοήθεια μιας υπηρεσίας τοποθεσίας (location service) και την περιλαμβάνει στην διεύθυνση προορισμού του πακέτου. Η

απόφαση δρομολόγησης σε κάθε κόμβο λαμβάνεται με βάση τη γεωγραφική θέση του κόμβου προορισμού που περιέχεται στο πακέτο και τις θέσεις των γειτόνων του κόμβου προώθησης. Έτσι, οι αλγόριθμοι της γεωγραφικής δρομολόγησης δεν απαιτούν ούτε την δημιουργία και διατήρηση μονοπατιών, ούτε την αποθήκευση πινάκων δρομολόγησης γλιτώνοντας έτσι και την αποστολή μηνυμάτων για να τους κρατάνε ενήμερους. Επίσης, πολύ σημαντικό είναι το γεγονός ότι η παράδοση των πακέτων δεν επηρεάζεται από αλλαγές στην τοπολογία του δικτύου ανάμεσα στον αποστολέα και τον παραλήπτη.

Τέλος, ένα άλλο πλεονέκτημα αυτής της προσέγγισης είναι ότι υποστηρίζει την παράδοση πακέτων σε όλους τους κόμβους μιας δοσμένης γεωγραφικής περιοχής. Αυτό το είδος υπηρεσίας ονομάζεται γεωεκπομπή (geocasting) και είναι ένα είδος πολυεκπομπής. Επίσης ένας κόμβος γνωρίζει μια πιο ακριβή διεύθυνση του προορισμού, έχει τη δυνατότητα να ενημερώσει τη διεύθυνση στο πακέτο πριν το προωθήσει. Όπως προαναφέραμε, απαιτείται η γνώση των θέσεων των γειτόνων του τρέχοντος κόμβου για να ληφθεί η απόφαση δρομολόγησης. Αυτό επιτυγχάνεται με την περιοδική αποστολή πλαισίων από κάθε κόμβο, που περιέχουν την διεύθυνσή τους. Με αυτό τον τρόπο οι κόμβοι ενημερώνονται για τους γείτονες τους.

4.4 Υπάρχοντα πρωτόκολλα δρομολόγησης

Στον παρακάτω πίνακα παρουσιάζονται τα πρωτόκολλα που έχουν αναπτυχθεί βάσει της προηγούμενης κατηγοριοποίησης

Κατηγορίες Πρωτοκόλλων	Πρωτόκολλα
<i>Δυναμικά</i>	• Babel
	• FSR
	• OLSR
	• DREAM
	• DSDV
<i>Αντιδραστικά</i>	• AODV
	• DSR
	• RDMAR
<i>Υβριδικά</i>	• ZRP
	• ToRA
	• AntHocNet
<i>Ιεραρχικά</i>	• HSR
	• VSR
	• CBRD
<i>Γεωγραφικά</i>	• Greedy - Άπληστος αλγόριθμος
	• LAR
	• FACE (GPSR)
	• GRID
	• ZHLS

Πίνακας 4.1 Υπάρχοντα πρωτόκολλα δρομολόγησης

4.5 Πρωτόκολλα δρομολόγησης βασισμένα στις αποικίες μυρμηγκιών

4.5.1 Πρωτόκολλο δρομολόγησης AntHocNet

Το πρωτόκολλο AntHocNet αποτελεί έναν αλγόριθμο δρομολόγησης που λειτουργεί εντελώς υβριδικά και προτάθηκε από τους Di Caro, Ducatelle και Gambardella το Σεπτέμβριο του 2004. Ανήκει στην ομάδα Ant Colony Optimization (ACO). Το AntHocNet είναι παράλληλα αντιδραστικό και δυναμικό πρωτόκολλο, έχοντας πολλές ομοιότητες με αντιδραστικά πρωτόκολλα όπως το AODV και δυναμικά πρωτόκολλα όπως το DSDV. Εντούτοις, καινούργιο γνώρισμα που χαρακτηρίζει το πρωτόκολλο αυτό είναι η συμπεριφορά που κατευθύνεται από πιθανότητες που αποθηκεύονται στους πινάκες φερομονών. Χρησιμοποιώντας τις πιθανότητες, τα πακέτα ελέγχου (στην περίπτωσή μας μυρμηγκία) επιτυγχάνουν να καταστήσουν το AntHocNet προσαρμοστικό και αυτό-διορθούμενο σε σχέση με τις απρόσμενες αλλαγές στην τοπολογία ενός MANET δικτύου. Τα γνωρίσματα της προσαρμοστικότητας και αυτό-διόρθωσης του AntHocNet ικανοποιούν κάποιες από τις απαιτήσεις που αφορούν τη βέλτιστη υβριδική συμπεριφορά ενός δικτύου.

Ο αλγόριθμος εκτελεί δύο φάσεις, την αντιδραστική και τη δυναμική, αφού αποτελεί μια συνένωση αντιδραστικών και δυναμικών χαρακτηριστικών:

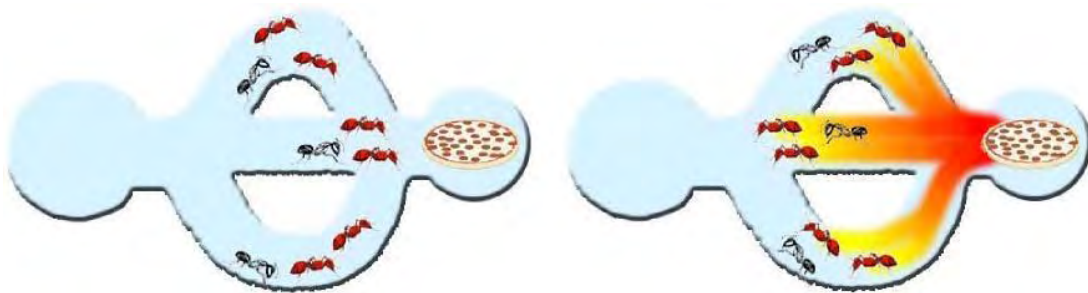
1. **Αντιδραστική Φάση:** Το αντιδραστικό στοιχείο του AntHocNet είναι υπεύθυνο για την εύρεση διαδρομής και τον καθορισμό μονοπατιού προς έναν άγνωστο προορισμό. Συνεπώς, όταν ο κόμβος-πηγή αρχίζει να επικοινωνεί με τον κόμβο-προορισμό και η διαδρομή προς τον κόμβο-προορισμό δεν είναι γνωστή, ο κόμβος - πηγή μεταδίδει ένα ορισμένο αριθμό αντιδραστικών μυρμηγκιών προώθησης (ΑΜΠ), για να βρει ένα μονοπάτι.
2. **Δυναμική Φάση:** Το δυναμικό μέρος του AntHocNet είναι υπεύθυνο για τη συντήρηση μονοπατιού και την εξερεύνηση με σκοπό την επίτευξη εναλλακτικών και καλύτερων διαδρομών προς τον κόμβο – προορισμό.

Το σημαντικότερο στοιχείο για την επιτυχία του δυναμικού μηχανισμού είναι οι πληροφορίες - δόλωμα που υπολογίζονται ως επί το πλείστον με τη χρήση μηνυμάτων “Hello” που μεταδίδονται περιοδικά μεταξύ γειτονικών κόμβων

4.5.2 Πρωτόκολλο δρομολόγησης Accelerated Ants

Ο Matsuo και ο Mori το 2001 περιέγραψαν το πρώτο σχήμα δρομολόγησης για ad hoc δίκτυα που βασίζεται στους ACO αλγορίθμους και ονομάζεται επιτάχυνση δρομολόγησης μυρμηγκιών (accelerated ants routing). Φαίνεται ότι είναι μια απλή προσαρμογή ενός ήδη γνωστού σχήματος για δίκτυα επικοινωνιών, με δύο επιπλέον προσθήκες. Ακολούθησαν την μέθοδο Ants-Routing και πρόσθεσαν έναν κανόνα “μη επιστροφής” όπου αποθαρρύνει τα μυρμηγκία να επιλέξουν την γειτονιά από την οποία προήλθε το μήνυμα . Επίσης πρόσθεσαν έναν κανόνα “Ν βημάτων για προς τα πίσω εξερεύνηση”. Ο αλγόριθμος αυτός χρησιμοποιείται όταν ένα μυρμηγκί κινείται προς τα πίσω (και ως εκ τούτου οι καταχωρήσεις

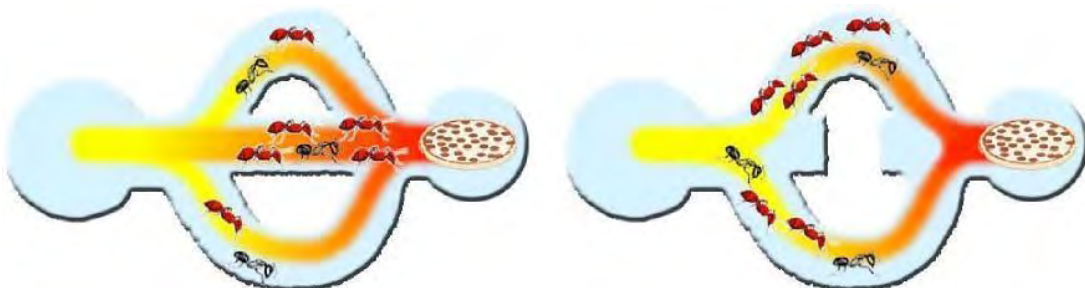
δρομολόγησης προς τον προορισμό είναι ενημερωμένες). Η αξιολόγηση των επιδόσεων έδειξε ότι ο αλγόριθμος δρομολόγησης των νέων μυρμηγκιών εμφανίζει μεγαλύτερη επιτάχυνση στη δρομολόγηση των πακέτων σε σχέση με την Ants-Routing μέθοδο, ακόμη και αν η τοπολογία του δικτύου είχε δυναμικά, αλλάξει. Το επιταχυνόμενο μοντέλο δρομολόγησης χρησιμοποιεί ομοιόμορφα πιθανοτικά μυρμηγκία. Τα ομοιόμορφα μυρμηγκία είναι απαραίτητα στα ad hoc δίκτυα λόγω των μη σταθερών δεσμών μεταξύ των κόμβων. Όταν μια διαδρομή που ήδη ήταν επιλεγμένη διαδρομή, χαλάσει, τότε τα ομοιόμορφα μυρμηγκία μπορούν εύκολα να βρουν μια εναλλακτική διαδρομή. Ο αλγόριθμος παρουσιάζεται στις παρακάτω εικόνες (εικόνα 4.2 α και β)



Εικόνα 4.2 α) Αναζήτηση τροφής

β) Η φερομόνη οδηγεί στην τροφή

Στην εικόνα 4.3α παρουσιάζονται τόσο τα πιθανοτικά μυρμηγκία (με κόκκινο) όσο και τα ομοιόμορφα (με μαύρο) που διαλέγουν με ομοιόμορφο τρόπο το μονοπάτι που θα ακολουθήσουν μιας και δεν υπάρχει κανένα ίχνος φερομόνης στο σύστημά μας. Στην εικόνα β παρουσιάζεται ο τρόπος με τον οποία τα μυρμηγκία αφήνουν την φερομόνη καθώς παίρνουν τον δρόμο επιστροφής για την φωλιά. Το μεσαίο μονοπάτι είναι το συντομότερο γι'αυτό και έχει και την μεγαλύτερη συγκέντρωση φερομόνης. Κατά συνέπεια, τα πιθανοτικά μυρμηγκία που φαίνονται στην παρακάτω εικόνα ακολουθούν αυτό το μονοπάτι. Η εικόνα 4.3β παρουσιάζει πως αν το μονοπάτι που ήδη είχαν επιλέξει ως το συντομότερο καταστραφεί, τότε τα μυρμηγκία θα προσαρμοστούν σε αυτή την αλλαγή και θα επιλέξουν το νέο συντομότερο μονοπάτι.



Εικόνα 4.3 α) Συντομότερη διαδρομή είναι πιο ενισχυμένη

β) Η σύνδεση χάνεται

4.5.3 Πρωτόκολλο δρομολόγησης Source Update

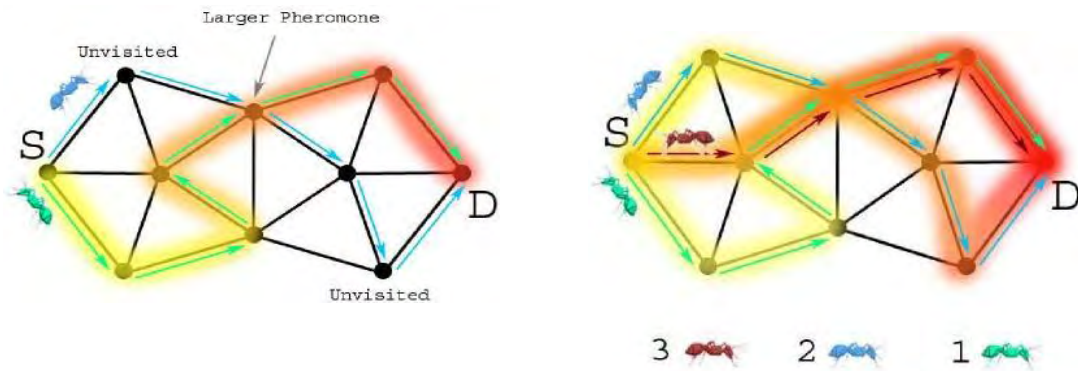
Οι Islam, Thulasiraman και Thulasiram πρότειναν έναν ACO αλγόριθμο που ονομάζεται “ενημέρωση πηγής” (source update routing), για όλα τα ζεύγη δρομολόγησης σε ad hoc δίκτυα. Με τον όρο “όλα τα ζεύγη δρομολόγησης” εννοούμε ότι δημιουργούνται πίνακες δρομολόγησης σε κάθε κόμβο, για όλα τα ζεύγη πηγής-προορισμού, με την μορφή ενός πίνακα που για γραμμές έχει όλους τους γειτονικούς κόμβους και για στήλες όλους τους πιθανούς προορισμούς.

Στην τεχνική ενημέρωσης πηγής, κάθε μυρμήγκι απομνημονεύει ολόκληρη τη διαδρομή προς τον προορισμό του και την χρησιμοποιεί για να επιστρέψει πίσω στην πηγή. Ενώ το μυρμήγκι βρίσκεται σε αναζήτηση του προορισμού, οι ενημερώσεις του πίνακα δρομολόγησης εκτελούνται με στόχο τον σχηματισμό ενός μονοπατιού που οδηγεί πίσω στην πηγή. Κατά τη διάρκεια της κίνησης προς τα πίσω, οι ενημερώσεις γίνονται με σεβασμό προς τον επιλεγμένο προορισμό D (όπου D είναι το αρχικό σημείο στη διαδρομή, διαγράφοντας έτσι το συσσωρευμένο βάρος πρώτα), ο οποίος χρησιμεύει ως πηγή του νέου μηνύματος, επομένως, η διαδικασία για την προς τα πίσω κίνηση είναι αλγοριθμικά ταυτόσημη με εκείνη που χρησιμοποιείται στην προς τα εμπρός κίνηση. Η δρομολόγηση προς τα πίσω απαιτείται έτσι ώστε το S να τοποθετήσει τελικά κάποια φερομόνη στον πίνακα δρομολόγησης του για D. Το ποσό της φερομόνης που εναποτίθεται σε κάθε επιλεγείσα ακμή δεν είναι σταθερό. Εξαρτάται από τον χρόνο μετάδοσης (καθυστέρησης), την συμφόρηση, τον χρόνο αλληλεπίδρασης ή από άλλες μετρήσεις. Επισημαίνεται ότι το ποσό της νέας φερομόνης που απομένει στον ήδη επισκεπτόμενο σύνδεσμο είναι αντιστρόφως ανάλογος με το αθροιστικό βάρος από το S ως τον τρέχοντα κόμβο, έτσι ώστε οι μακρύτερες διαδρομές να είναι λιγότερο πιθανό να ακολουθηθούν.

Για να αποθηκεύσουν τη διαδρομή, τα μυρμήγκια χρησιμοποιούν μια στοίβα που περιέχει όλους τους κόμβους κατά μήκος της διαδρομής από το S. Η ίδια στοίβα χρησιμοποιείται για την ανίχνευση και αποφυγή βρόχων. Αυτό επιτυγχάνεται αγνοώντας γείτονες που είναι ήδη στη στοίβα, όταν αποφασίζει να μεταβεί στο επόμενο hop. Συνεπώς, ποτέ δεν δημιουργείται βρόχος. Εάν ένας κόμβος δεν έχει κανένα γείτονα που να μην είναι ήδη στη στοίβα η αναζήτηση οπισθοχωρεί στον προηγούμενο κόμβο.

Τα διερευνητικά μυρμήγκια ακολουθούν ημι – ντετερμινιστικό σύστημα όταν θέλουν να αποφασίσουν τον επόμενο κόμβο που θα επισκεφτούν, κάνοντας αναζήτηση κατά βάθος. Εάν υπάρχει οποιαδήποτε σύνδεση προς κάποιο αθέατο γείτονα (κόμβος που δεν ανήκει ούτε στην στοίβα ούτε έχει επισκεφθεί από κάποιο μυρμήγκι προηγουμένως) που δεν έχει ακόμη δοκιμαστεί από οποιοδήποτε άλλο μυρμήγκι, τότε επιλέγεται αυτή η σύνδεση . Και αυτό γιατί πρέπει να δοκιμάσουμε την ποιότητα όλων των υποψήφιων πιθανών διαδρομών. Η προηγούμενη δοκιμασία είναι άκρως απαραίτητη μιας και η τοπολογία των ad-hoc δικτύων συνεχώς μεταβάλλεται και αυτό μπορεί να έχει σαν αποτέλεσμα να δημιουργηθεί κάποιο καλύτερο μονοπάτι. Εάν δεν υπάρχει κάποιος αθέατος κόμβος, τότε τα μυρμήγκια σαν επόμενο προς εξέταση κόμβο επιλέγουν αυτόν, του οποίου το μονοπάτι φέρει

μεγαλύτερες συγκεντρώσεις φερομόνης. Επιλέγει δηλαδή, τον γείτονα του οποίου η τιμή του ίχνους φερομόνης στη στήλη που αντιστοιχεί στον προορισμό D, είναι η μεγαλύτερη.



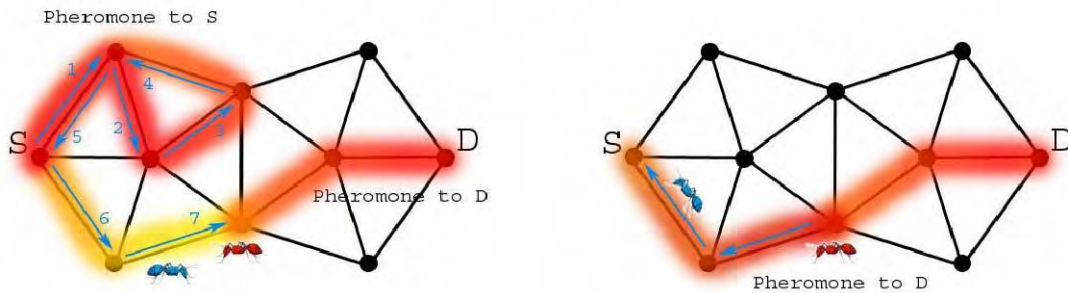
Εικόνα 4.4 Αλγόριθμος δρομολόγησης Source Update

Στην Εικόνα 4.4 απεικονίζεται αλγόριθμος δρομολόγησης Source Update. Δείχνει πώς τα μυρμήγκια προτιμούν να επισκεφτούν κόμβους που δεν έχουν ξαναεπισκευτεί. Αν δεν υπάρχουν αθέατοι κόμβοι τότε θα επιλέξουν τον κόμβο με την υψηλότερη συγκέντρωση της φερομόνης. Τα βέλη στα δύο σχήματα απεικονίζουν την προς τα εμπρός κίνηση των μυρμηγκιών, και το χρωματισμένο μονοπάτι με την συγκέντρωση φερομόνης απεικονίζει την προς τα πίσω κίνηση. Το καφέ μυρμήγκι ξεκινάει τελευταίο την αναζήτηση του προορισμού D και έτσι ανακαλύπτει μια συντομότερη διαδρομή από αυτή των προκατόχων του την οποία και τελικά ακολουθεί.

4.5.4 Πρωτόκολλο δρομολόγησης Random Walk

Στον αλγόριθμο δρομολόγησης Random Walk κάθε μυρμήγκι εκτελεί μια τυχαία βόλτα πάνω από το δίκτυο. Σε αυτό τον τυχαίο περίπατο μυρμήγκια και ομοιόμορφα πακέτα επιλέγουν τυχαία τον επόμενο hop τους. Η πηγή (S) στέλνει πολλά πακέτα μυρμηγκιών στην προσπάθεια να βρεθεί διαδρομή προς τον προορισμό (D). Κατά τη διάρκεια της τυχαίας βόλτας αφήνουν μονοπάτια φερομόνης. Τα μυρμήγκια δεν ψάχνουν για κάποια συγκεκριμένη διαδρομή απλά για κάποιο μονοπάτι που φέρει ίχνη φερομόνης. Όταν ένα μυρμήγκι φτάνει σε έναν κόμβο που περιέχει φερομόνη τότε στέλνει ένα πακέτο απάντησης στην πηγή. Αυτό το πακέτο απάντησης προτού φτάσει στην πηγή θα ενημερώσει και τους ενδιαμέσους κόμβους για τον κόμβο που βρέθηκε. Παράλληλα πάνω από το δίκτυο στέλνονται πακέτα “Hello” προκειμένου να ανακαλυφθούν ποιοί κόμβοι είναι ενεργοί ακόμα ή ποιοι έχουν απομονωθεί από το δίκτυο. Κάθε κόμβος από την άλλη, διαχέει δυναμικά πακέτα, σε όλο το δίκτυο προκειμένου να διαφημίσει την ύπαρξή του και αυτό είναι εξαιρετικά χρήσιμο μιας και έτσι μπορεί να μειωθεί η κίνηση των αιτήσεων πάνω από το δίκτυο. Τέλος χρησιμοποιείται το πεδίο time to

live του πίνακα δρομολόγησης με αποτέλεσμα την διάδοση των “κακών”/χρονοβόρων διαδρομών για να αποφεύγονται μελλοντικά.



Εικόνα 4.5 Αλγόριθμος δρομολόγησης Random Walk

Τα μυρμήγκια μπορούν να επωφεληθούν του ασύρματου μέσου μετάδοσης, επειδή είναι δυνατόν να ακούσουν όλες τις εκπομπές για κάθε κόμβο. Πληροφόρηση για την δρομολόγηση μπορούμε να έχουμε ακούγοντας όλη την κυκλοφορία πάνω από το δίκτυο. Έτσι εύκολα και γρήγορα μπορούν να ανιχνευτούν νέοι κόμβοι ακούγοντας τις μεταδόσεις τους. Επίσης, ένα μεγάλο μέρος πληροφοριών σχετικά με το δίκτυο μπορεί να αποκτηθεί από τους προορισμούς που οι γειτονικοί κόμβοι προωθούν.

Το σχήμα απεικονίζει τον αλγόριθμο δρομολόγησης Random Walk. Το κόκκινο μυρμήγκι στο αριστερό σχήμα έχει αφήσει ένα ίχνος φερομόνης από την τρέχουσα θέση του μέχρι τον προορισμό D. Το μπλε μυρμήγκι κινείται με τυχαίο τρόπο (επισημαίνονται με τα αριθμημένα μπλε βέλη) κατά μήκος του δικτύου μέχρι που ανιχνεύει την φερομόνη που έχει αφήσει το κόκκινο μυρμήγκι το οποίο επιστρέφει στην πηγή (S) έχοντας βρει τον τελικό προορισμό (D). Καθώς ψάχνει για ένα μονοπάτι προς τον προορισμό, αφήνει ένα μονοπάτι φερομόνης που οδηγεί πίσω στην πηγή. Αποδεικνύεται τότε ότι αρχίζει να αφήνει και ένα δεύτερο μονοπάτι φερομόνης όπως φαίνεται στο δεξί σχήμα. Με αυτό τον τρόπο σχηματίζεται το τελικό μονοπάτι που οδηγεί από την πηγή στον προορισμό.

4.5.5 Πρωτόκολλο δρομολόγησης PERA

Στον πιθανοτικά αναδυόμενο αλγόριθμο (Probabilistic Emergent Routing Algorithm) η ανακάλυψη διαδρομής γίνεται μέσα από τις πλημμύρες του δικτύου με μυρμηγκία. Κατά την διάρκεια της πλημμύρας, τα μυρμηγκία εκτελούν κινήσεις προς τα εμπρός και προς τα πίσω και γεμίζουν τους πίνακες δρομολόγησης με πιθανότητες. Αυτές οι πιθανότητες αντανακλούν αν ένας γείτονας έχει δυνατότητα να διαβιβάσει ένα πακέτο για έναν συγκεκριμένο προορισμό.

Αρχικά κάθε κόμβος μπορεί να ανακαλύψει τους γείτονές του ανταλλάσσοντας πακέτα μηνυμάτων “Hello”. Τα αποτελέσματα, όμως της ανταλλαγής μηνυμάτων εκχωρούνται στον πίνακα δρομολόγησης αφού πρώτα φτάσει ένα μυρμηγκί το οποίο έχει βρει τον τελικό προορισμό. Κάθε μονοπάτι από τα πιθανά μονοπάτια που οδηγούν προς τον προορισμό έχει την ίδια πιθανότητα να επιλεγεί. Η πιθανότητα όμως αυτή, αυξάνεται για το μονοπάτι που ακολουθεί το μυρμηγκί που εκτελεί κίνηση προς τα πίσω (από τον προορισμό στην πηγή) μιας και αφήνει πίσω του φερομόνη.

Καθώς μυρμηγκία πλημμυρίζουν το δίκτυο ο αλγόριθμος χρησιμοποιεί σειριακούς αριθμούς ώστε να αποφευχθεί η ύπαρξη διπλών πακέτων. Μόνο ο μεγαλύτερος σειριακός αριθμός που έχουμε από το προηγούμενο hop προωθείται στον πίνακα δρομολόγησης. Τα μυρμηγκία που κινούνται προς τα εμπρός και έχουν μικρότερο σειριακό αριθμό απορρίπτονται και απομακρύνονται από το δίκτυο. Η προσέγγιση αυτή του PERA μοιάζει με τον τρόπο που αποστέλλει πακέτα αιτήσεων ο αλγόριθμος AODV με την διαφορά ότι ο πρώτος ανακαλύπτει ένα σύνολο πολλών πιθανών μονοπατιών έναντι το ενός μονοπατιού που ανακαλύπτει ο δεύτερος. Τα πακέτα δεδομένων μπορεί να δρομολογηθούν στο επόμενο hop βάσει της υψηλότερης πιθανότητας που υπάρχει στον πίνακα δρομολόγησης, ή με στοχαστικό τρόπο που αποδίδει καλύτερα σε σταθερά δίκτυα με μικρές τοπολογίες.

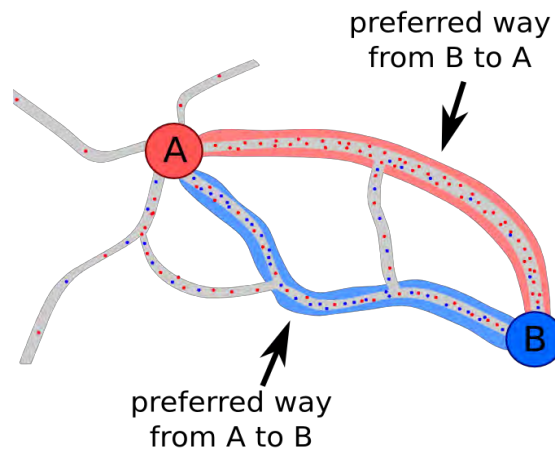
4.5.6 Πρωτόκολλο δρομολόγησης ARA

Ο αλγόριθμος Ant-Colony Based (ARA) δρομολόγησης αποτελείται από τρεις φάσεις. Η πρώτη αναφέρεται ως φάση Route Discovery, κατά την οποία ανακαλύπτονται νέα μονοπάτια. Η δημιουργία νέων διαδρομών απαιτεί τη χρήση ενός μυρμηγκιού που κινείται προς τα εμπρός (forward ant – FANT) και καθορίζει τη φερομόνη που οδηγεί στον κόμβο-πηγή και ενός μυρμηγκιού που κινείται προς τα πίσω (backward ant – BANT) που καθορίζει τη διαδρομή για το κόμβο-προορισμό. Τα FANTS μεταδίδουν πακέτα από τον αποστολέα προς όλους τους γείτονές του. Κάθε Fant έχει έναν σειριακό αριθμό προκειμένου να αποφευχθούν τα διπλότυπα.

Ένας κόμβος που λαμβάνει FANT για πρώτη φορά, δημιουργεί ένα αρχείο (διεύθυνση προορισμού, επόμενο hop, αξία φερομόνης) στον πίνακα δρομολόγησης του. Ο κόμβος ερμηνεύει τη διεύθυνση της πηγής του FANT ως διεύθυνση προορισμού, την διεύθυνση του προηγούμενου κόμβου ως το επόμενο hop που θα επισκεφτεί, και υπολογίζει τη φερομόνη βάσει του αριθμού των hops που χρειάζεται για φτάσει στον προορισμό του. Έπειτα ο κόμβος αναμεταδίδει το FANT στους γείτονές του. Όταν το FANT φτάσει στον τελικό προορισμό, υποβάλλεται σε

επεξεργασία και αφού ο κόμβος προορισμού εξάγει τις πληροφορίες που επιθυμεί κατόπιν καταστρέφεται. Ένα μυρμήγκι BANT δημιουργείται και αποστέλλεται προς τον κόμβο της πηγής. Με αυτόν τον τρόπο, καθορίζεται η διαδρομή και τα πακέτα δεδομένων που μπορούν να σταλούν.

Στη δεύτερη φάση, που ονομάζεται Route Maintenance, οι διαδρομές που έχουν βρεθεί βελτιώνονται κατά τη διάρκεια της επικοινωνίας στο δίκτυο. Πακέτα δεδομένων χρησιμοποιούνται για την διατήρηση, την ενίσχυση του βέλτιστου μονοπατιού καθώς και τον περιορισμό το μεγάλου ποσού επιπλέον πληροφορίας (overhead) στο δίκτυο. Οι τιμές της φερομόνης αλλάζουν. Όταν ένας κόμβος υί στέλνει ένα πακέτο δεδομένων με στόχο να φτάσει τον κόμβο-προορισμό υD, περνάει πρώτα από τον γειτονικό του κόμβο αυξάνοντας την τιμή της φερομόνης στο $(υί, υD, ψ)$ δψ. Το ίδιο συμβαίνει και προς την αντίθετη κατεύθυνση. Η διαδικασία εξάτμισης της φερομόνης προσομοιώνεται από την τακτική μείωση των τιμών φερομόνης.



Εικόνα 4.6 Με μπλε σημειώνεται η διαδρομή των FANTS με κόκκινο των BANTS

Η τρίτη φάση χειρίζεται τις αποτυχίες που προκύπτουν κατά την δρομολόγηση, ιδίως λόγω της κινητικότητας των κόμβων, ένα κοινό πρόβλημα που αντιμετωπίζουμε στα δίκτυα MANETS. Το ARA αναγνωρίζει την αποτυχία μιας διαδρομής μέσα από ένα χαμένο ACK (τμήμα γνωστοποίησης). Οι συνδέσεις απενεργοποιούνται θέτοντας την τιμή της φερομόνης σε 0 και τότε ο κόμβος αναζητά κάποια εναλλακτική σύνδεση. Εάν υπάρχει μια τέτοια διαδρομή τότε επιλέγεται αυτή, αλλιώς όλοι οι κόμβοι ενημερώνονται για την αποτυχία.

4.5.7 Πρωτόκολλο δρομολόγησης AntAODV

Οι Marwaha, Tham και Srinivasan μελέτησαν μια υβριδική προσέγγιση χρησιμοποιώντας τόσο το πρωτόκολλο AODV όσο και τον τρόπο εξερεύνησης των αντιδραστικών μυρμηγκιών. Η τεχνική τους ονομάζεται AntAODV. Οι πίνακες δρομολόγησης στον AntAODV και στον AODV είναι ίδιοι. Αν ο κόμβος-αποστολέας, δηλαδή η πηγή S διαθέτει μια διαδρομή προς τον προορισμό, τότε την χρησιμοποιεί για να προωθήσει το πακέτο. Κάθε μυρμήγκι ακολουθεί μια προσέγγιση πλημμυρίδας και δημιουργεί πολλά αντίγραφα του εαυτού του. Αν ένα μυρμήγκι ανακαλύψει μια νέα καλύτερη διαδρομή τότε επιστρέφει αμέσως στην πηγή S για να την αναφέρει. Η βασική διαφορά του AntAODV με τον AODV τελικά, είναι πως ο πρώτος έχει μεγαλύτερο πίνακα δρομολόγησης που περιέχει όλους τους γείτονες και την ποσότητα της φερομόνης που υπάρχει στο μεταξύ τους μονοπάτι, έναντι της μίας καλύτερης επιλογής που κρατάει ο δεύτερος στον πίνακα δρομολόγησης του. Αυτό επιτρέπει μια τυχαία επιλογή για την μετάβαση στο επόμενο hop.

Κεφάλαιο 5ο

5 Αλγόριθμος ACO σε Clouds

5.1 Εισαγωγή

Το Cloud Computing αποτελεί μια νέα προσέγγιση στον χώρο των καταναμημένων συστημάτων που χρησιμοποιεί και κάποιες υφιστάμενες τεχνολογίες. Σκοπός του είναι η παροχή πόρων με την μορφή υπηρεσίας στους χρήστες του συστήματος. Τέτοιοι πόροι είναι η υπολογιστική ισχύ (CPU), η αποθηκευτική δυνατότητα (storage) κ.ά. Βέβαια Cloud Computing δεν μπορεί να είναι το κάθε σύστημα το οποίο αναθέτει υπολογιστικές και αποθηκευτικές υπηρεσίες εξωτερικά.

Ένας τεχνικός ορισμός του cloud computing το ορίζει ως «υπολογιστική ικανότητα βασισμένη στο διαδίκτυο» που παρέχεται ως μια αφηρημένη έννοια ανάμεσα στους υπολογιστικούς πόρους και την υποκείμενη τεχνική αρχιτεκτονική που την υποστηρίζει (π.χ. διακομιστές, συστήματα αποθήκευσης, δίκτυα). Το cloud computing είναι ένα μοντέλο που επιτρέπει μια άνετη on-demand πρόσβαση δικτύου σε ένα κοινό σημείο συγκέντρωσης διαμορφώσιμων υπολογιστικών πόρων οι οποίοι μπορούν να τροφοδοτηθούν άμεσα και να ελευθερωθούν με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδραση από υπηρεσία παρόχου». Αυτός ο ορισμός αναφέρει ότι τα «clouds» έχουν πέντε βασικά χαρακτηριστικά: on-demand αυτοεξυπηρέτηση, ευρεία πρόσβαση στο δίκτυο, συγκέντρωση των πόρων, ταχεία ελαστικότητα, και μετρήσιμη υπηρεσία.



Εικόνα 5.1 Δίκτυο Cloud

Οι ερευνητές τονίζουν ότι έτσι ένα «cloud» είναι ως εκ τούτου όχι μόνο ένας συνδυασμός συμπλεγμάτων και πλεγμάτων, αλλά επεκτείνεται επίσης με τη σιωπηρή χρήση των τεχνολογιών εικονοποίησης, όπως οι Εικονικές Μηχανές, Virtual Machines (VMs), για να ανταποκριθεί σε ένα ειδικά διαπραγματεύσιμο

επίπεδο ποιότητας υπηρεσία. Αυτός ο ορισμός συνεπάγεται και συλλαμβάνει δύο ενδεχόμενα προβληματικά ζητήματα: α) το ζήτημα των επιχειρήσεων για τη διαπραγμάτευση της σωστής SLA (Service Level Agreement) από την πλευρά του πελάτη και β) έχοντας την τεχνική ικανότητα, να μπορεί να υπολογίζει σωστά και να εγγυάται την περιγραφόμενη υπηρεσία στην SLA εξ ολοκλήρου (resource monitoring, failure redundancy, rebalancing of workloads, κ.α. από την οπτική του παρόχου). Ως εκ τούτου, ο ορισμός του GRIDS φαίνεται κάπως πιο ουδέτερος από αυτόν του Berkeley:

«To cloud computing αναφέρεται τόσο στις εφαρμογές που παρέχονται ως υπηρεσίες μέσω του Internet όσο και στο υλικό και τα συστήματα λογισμικού που υπάρχουν μέσα στα κέντρα δεδομένων που παρέχουν αυτές τις υπηρεσίες (Software as a Service - SaaS). Το υλικό και το λογισμικό του κέντρου δεδομένων είναι αυτό που θα ονομάσουμε Cloud. Όταν ένα Cloud γίνεται διαθέσιμο με τρόπο pay-as-you-go στο κοινό, το ονομάζουμε Public Cloud και η υπηρεσία που πωλείται είναι η Utility Computing»

Το cloud computing γενικά λειτουργεί πάνω στην ιδέα ότι, οι εργασίες που γίνονται στον υπολογιστή από τη μεριά του πελάτη μπορούν να μετακινηθούν πάνω σε ένα αόρατο σύμπλεγμα πόρων στο διαδίκτυο. Το cloud-computing είναι η ανάπτυξη και η χρήση της τεχνολογίας των υπολογιστών («computing»), βασισμένη στο διαδίκτυο («cloud»). Ως έννοια, είναι μια νέα προσέγγιση για την επιστήμη. Σύμφωνα με αυτή, τα στοιχεία αντλούνται από τους χρήστες, οι οποίοι πλέον δεν χρειάζεται να έχουν τη γνώση, την εμπειρία, ή τον έλεγχο της υποδομής της τεχνολογίας («in the cloud»), που τους υποστηρίζει.

5.2 Τύποι του cloud computing

Επί του παρόντος, οι προγραμματιστές εργάζονται με τρεις μεγάλες κατηγορίες «cloud» συστημάτων:

- **software-as-a-service**
- **platform-as-a-service**
- **infrastructure-as-a-service**

Οι τύποι του cloud computing:

5.2.1 SaaS (software-as-a-service)

Στο μοντέλο SaaS, ο χρήστης αγοράζει μια συνδρομή σε κάποιο προϊόν λογισμικού. Ωστόσο μερικά ή όλα τα δεδομένα και ο κώδικας βρίσκονται μακριά από αυτόν. Για παράδειγμα, το «Google Docs» προσφέρουν μια εναλλακτική λύση για το Microsoft Office καθώς αποθηκεύει έγγραφα στο server του Google. Δεν διατηρούν κάποιον κώδικα στο μηχάνημα του client, έστω κι αν ορισμένοι κώδικες ίσως εκτελεστούν στον client προσωρινά. Για παράδειγμα, το «Google Docs» στηρίζεται σε JavaScript, η οποία εκτελείται στο πρόγραμμα περιήγησης του Web, (Web browser). Σε αυτό το μοντέλο, οι εφαρμογές θα μπορούσαν να εκτελεστούν εξ

ολοκλήρου από το δίκτυο, με το περιβάλλον εργασίας του χρήστη να βασίζεται σε ένα «thin client».

5.2.2 Paas (platform-as-a-service)

Από την άποψη του καταναλωτή, το PaaS λογισμικό μοιάζει μάλλον με το SaaS, αλλά αντί οι προγραμματιστές λογισμικού να κατασκευάζουν το πρόγραμμα να εκτελείται σε δική τους υποδομή στο Web, το κάνουν ώστε να εκτελείται πάνω σε κάποιου άλλου την υποδομή. Αποτελεί ένα τρόπο δημιουργίας εφαρμογών που φιλοξενούνται σε ένα «cloud» πάροχο. Επιτρέπει στο χρήστη να εγκαταστήσει εφαρμογές χωρίς να χρειάζεται να δαπανήσει χρήματα για την αγορά διακομιστών που θα χρειαζόταν για να τις «φιλοξενήσει». Για παράδειγμα, η Google προσφέρει τη «Google App Engine», μια υπηρεσία που επιτρέπει σε οργανώσεις ανάπτυξης προγραμμάτων γραφής να εκτελούνται ειδικά στις υποδομές της Google. Το πιο γνωστό παράδειγμα εκτός από τη «Google AppEngine» είναι το «Windows Azure»

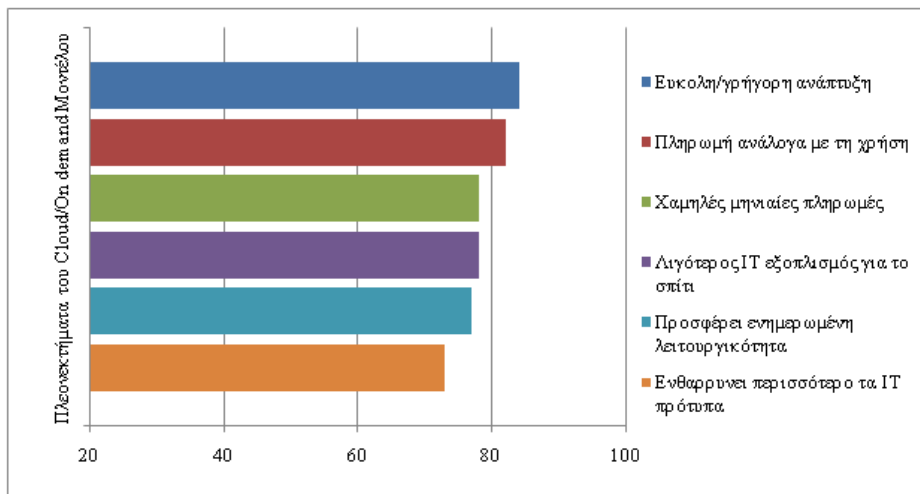
5.2.3 IaaS (infrastructure-as-a-service)

Παρόμοια με την PaaS, η IaaS επιτρέπει στην οργάνωση ανάπτυξης να ορίσει το δικό της λογισμικό περιβάλλον. Αυτό το περιβάλλον βασικά μεταφέρει τις εικόνες εικονικών διακομιστών (VM) στον πάροχο IaaS, αντί για προγράμματα, και αυτοί οι διακομιστές περιέχουν οτιδήποτε χρειάζονται οι προγραμματιστές. Ο πάροχος μπορεί αυτόματα να αυξήσει ή να συρρικνώσει τον αριθμό των εικονικών διακομιστών που λειτουργούν κάθε δεδομένη στιγμή. Έτσι τα προγράμματα μπορούν πιο εύκολα να επεξεργάζονται μεγάλα φορτία εργασίας εξοικονομώντας χρήματα, εφόσον δεν θα χρειάζονται πολλοί πόροι.

5.3 Χαρακτηριστικά του Cloud Computing

Στην ιδέα του «cloud computing» οι επιχειρήσεις βρίσκουν μια σημαντική νέα αξία στις υπηρεσίες του. Τα χαρακτηριστικά του «cloud» μπορούν να εξαλείψουν πολλούς από τους πολύπλοκους περιορισμούς του παραδοσιακού περιβάλλοντος των υπολογιστών, περιλαμβάνοντας χώρο, χρόνο, δύναμη, και κόστος. Οι υπηρεσίες του «Cloud», όπως τα κοινωνικά δίκτυα και τα συνεργατικά εργαλεία (βιντεοδιάσκεψη, διαχείριση εγγράφων, και webinars) αλλάζουν τον τρόπο με τον οποίο οι άνθρωποι στις επιχειρήσεις έχουν πρόσβαση στην πληροφορία, την διατυπώνουν, και την κατανοούν. Οι υποδομές και τα χαρακτηριστικά του «cloud computing» κάνουν κάτι τέτοιο να διευκολύνει τις εταιρείες να διαχειριστούν τα υπολογιστικά τους συστήματα ως ένα σημείο συγκέντρωσης πόρων περισσότερο παρά ως ένα σύνολο ανεξάρτητων περιβαλλόντων. Τα χαρακτηριστικά του «cloud computing» κρύβουν ορισμένες από τις θεμελιώδεις έννοιες του. Παρακάτω αναλύονται τα κυριότερα:

- **Εξελισμότητα/Ελαστικότητα (Scalability/Elasticity):** Η ικανότητα που έχουν τα υπολογιστικά συστήματα του «cloud computing» να αναπτύσσονται και να αυξάνονται σε ισχύ ανάλογα με την αύξηση της ζήτησης και των αναγκών του πελάτη. Σε αυτό το χαρακτηριστικό έχουν στηριχτεί όλες οι δομές και οι πλατφόρμες του cloud.
- **Διαθεσιμότητα (Availability):** Τα συστήματα του cloud computing είναι σχεδιασμένα με τέτοιο τρόπο, ώστε ακόμα και αν αποτύχει ένα τμήμα ή ένα στοιχείο τους, αυτό να μην συνεπάγεται την κατάρρευση ολόκληρου του συστήματος. Αυτό το χαρακτηριστικό είναι ιδιαίτερα σημαντικό αν σκεφτούμε πως όλα τα δεδομένα βρίσκονται αποθηκευμένα στο cloud, οπότε οι πιθανότητες κατάρρευσής του θα πρέπει να μειωθούν στο ελάχιστο δυνατό.
- **Ανεξαρτησία συσκευής, τοποθεσίας (Device & location independence):** Οι εφαρμογές αλλά και τα συστήματα που αναπτύσσονται στο cloud θα πρέπει να είναι ανεξάρτητα από τη συσκευή στην οποία τρέχουν, από το λειτουργικό που αυτή χρησιμοποιεί και από την τοποθεσία στην οποία βρίσκεται. Μόνον αν ισχύουν αυτά τα χαρακτηριστικά, μπορούν οι εφαρμογές του «cloud computing» να γίνουν κοινώς αποδεκτές.
- **Οικονομία (Economy):** Ένα άλλο κοινό χαρακτηριστικό που έχουν και οι υπηρεσίες του «cloud» είναι ότι διατηρούν το κόστος των πληροφοριακών συστημάτων σε πολύ χαμηλό επίπεδο. Ας μην ξεχνάμε πως ένας από τους λόγους για τους οποίους οι εταιρείες στρέφονται στη λύση του «cloud computing» είναι τα οικονομικά συμφέροντα που υπάρχουν.



Εικόνα 5.2 Πλεονεκτήματα Cloud Computing

5.4 Χρήση ACO αλγορίθμων σε Clouds

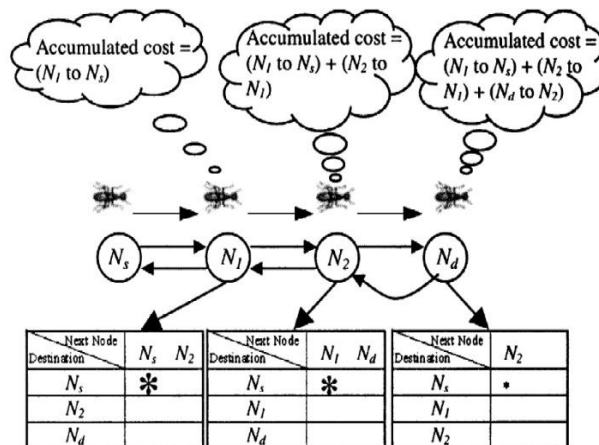
5.4.1 Χρήση του Aco για load balancing στα Clouds

Το πρόβλημα της εξισορρόπησης φόρτου στα Clouds αποτελεί πρόκληση για τους ερευνητές δικτύων. Έχοντας ένα ισορροπημένο φόρτο μπορούμε να καταφέρουμε να βελτιώσουμε σημαντικά την απόδοση, να έχουμε ένα εφεδρικό πλάνο σε περίπτωση που το σύστημά μας αποτύχει έστω και μερικώς, να διατηρήσουμε το σύστημα σε μία σταθερή κατάσταση και επίσης να διευκολύνουμε την τυχόν μελλοντική τροποποίηση του συστήματός μας.

Τον Απρίλιο του 2012 οι Ratan Mishra και Anant Jaiswal πρότειναν την χρήση του αλγορίθμου ACO ως τρόπο αντιμετώπισης του προβλήματος σε περιβάλλοντα Cloud. Κατασκεύασαν ένα σύστημα ελέγχου βασισμένο στα μυρμήγκια. Κάθε κόμβος στο δίκτυο ρυθμίστηκε ώστε

- να έχει συγκεκριμένη χωρητικότητα
- να υπάρχει πιθανότητα ο ίδιος να είναι ο κόμβος προορισμού
- να διαθέτει πίνακα φερομόνης

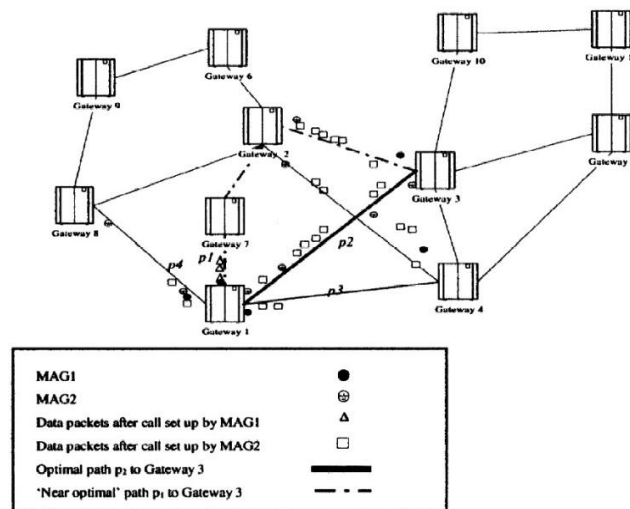
κάθε γραμμή στον πίνακα φερομόνης αντιπροσωπεύει την καλύτερη επιλογή για κάθε προορισμό και κάθε στήλη αντιπροσωπεύει την πιθανότητα να διαλέξει κάποιον γειτονικό του κόμβου ως επόμενο hop. Τα μυρμήγκια ξεκινάνε από τον αρχικό κόμβο και κινούνται προς τυχαίους προορισμούς. Σε αυτή την προσέγγιση τα μυρμήγκια που φτάνουν σε έναν κόμβο είναι υπεύθυνα να ενημερώσουν τον πίνακα με τις φερομόνες.



Εικόνα 5.2 Πίνακας φερομονών

Αν για παράδειγμα θέλουμε να δημιουργήσουμε μια σύνδεση μεταξύ 2 gateways σε ένα circuit – switched δίκτυο χρησιμοποιώντας τον αλγόριθμο Aco τότε αυτό θα γίνει όπως παρουσιάζεται παρακάτω. Θα δημιουργηθούν δύο ομάδες από κινητούς πράκτορες που θα αντιστοιχούν σε μία αποικία μυρμηγκιών η κάθε μία και

ο πίνακας δρομολόγησης της κάθε ομάδας αντιστοιχεί στον πίνακα φερομόνης κάθε αποικίας. Ακόμα κι αν οι δύο ομάδες των κινητών πρακτόρων (MAG1 και MAG2) μπορεί να έχουν τις δικές τους προτιμήσεις δρομολόγησης, μπορούν να λαμβάνουν υπόψη τους και τις προτιμήσεις της δρομολόγησης της άλλης ομάδας. Καθώς οι προτιμήσεις δρομολόγησης των μυρμηγκιών καταγράφονται στους πίνακες φερομόνης, οι προτιμήσεις δρομολόγησης των κινητών πρακτόρων αποθηκεύονται στους πίνακες δρομολόγησής τους.



Εικόνα 5.3 Σύνδεση μεταξύ 2 Gateways

Για την κατασκευή του πίνακα δρομολόγησης, η ομάδα πρακτόρων MAG1 (αντίστοιχα, MAG2) συμβουλευτεί τον πίνακα δρομολόγησης της MAG2 (αντίστοιχα, MAG1), ώστε να αποφευχθεί η δρομολόγηση πακέτων στις διαδρομές, που προτιμάει η άλλη ομάδα. Με αυτόν τον τρόπο αυξάνει την πιθανότητα ότι θα δημιουργηθούν δύο διαφορετικές συνδέσεις μεταξύ των πυλών πράγμα που σημαίνει πως έχουμε μια κατανεμημένη κίνηση πάνω στο Cloud. Καθώς ένας Web Server αυξάνει ή μειώνει τις απαιτήσεις του, οι αιτούμενες υπηρεσίες εκχωρούνται δυναμικά ώστε να ανταποκριθεί καλύτερα το σύστημά στις αλλαγές του χρήστη. Οι διακομιστές ομαδοποιούνται κάτω από έναν εικονικό διακομιστή ο καθένας από τους οποίους έχει την δική του εικονική ουρά που αποθηκεύονται οι υπηρεσίες.

Κάθε διακομιστής επεξεργάζεται μια αίτηση που λαμβάνει από την ουρά που υπολογίζει το κέρδος, το οποίο εξαρτάται από την ποσότητα της φερομόνης που έχουν αφήσει τα μυρμηγκία. Κάθε διακομιστής έχει αναλογικά έναν κινητό πράκτορα. Ο διακομιστής μετά την επεξεργασία του αιτήματος, δημοσιεύει τα κέρδη του πίνακα φερομόνης ο οποίος διατηρείται σε όλους του διακομιστές. Ένας server μπορεί να επιλέξει μια ουρά από έναν εικονικό διακομιστή με πιθανότητα p_x . Ένας server εξυπηρετεί μια αίτηση, υπολογίζοντας το κέρδος και συγκρίνοντάς το με το κέρδος της αποικίας και στη συνέχεια ρυθμίζει την πιθανότητα p_x . Εάν το κέρδος είναι υψηλό, τότε ο διακομιστής παραμένει στον τρέχοντα διακομιστή και διαφημίζει την απόφασή του στο Cloud. Αν το κέρδος του όμως είναι μικρό τότε παραμένει στον προηγούμενο διακομιστή.

5.4.2 Χρήση του Aco για τοποθέτηση φόρτου εργασίας σε Clouds με περιορισμένη κατανάλωση ενέργειας

Οι Eugen Feller, Louis Rillingy και Christine Morin το 2011 διατύπωσαν έναν αλγόριθμο για την κατανομή του φόρτου εργασίας ενός Cloud ο οποίος είναι ενεργειακά φιλικό προς το περιβάλλον.

Βασικά στοιχεία του αλγορίθμου είναι τα παρακάτω:

1. Ο αλγόριθμος παίρνει ως είσοδο ένα σύνολο των στοιχείων (items) και ένα σύνολο κάδων (bins), συμπεριλαμβανόμενο και του αντίστοιχα χρονικά μεταβαλλόμενο διάλυσμα πόρων. Οι παράμετροι αρχικοποιούνται και τα μονοπάτια φερομόνης λαμβάνουν την max τιμή (γραμμή 4).
2. Ο αλγόριθμος επαναλαμβάνεται μέχρι να φτάσει τον nCycles καθορισμένο αριθμό επαναλήψεων (γραμμές 5 έως 35). Σε κάθε επανάληψη, κάθε ένα μυρμήγκι ένα ανοίγει ένα κάδο v και αρχίζει να κατασκευάζει μια λύση S_a (γραμμές 6 έως 20). Αυτό επιτυγχάνεται με την πρώτη αρχικοποίηση του συνόλου των στοιχείων IS, των στοιχείων δηλαδή του δυαδικού πίνακα λύσεων
3. Ο αλγόριθμος εισέρχεται σε ένα βρόχο και ξεκινά την ανάθεση των αντικειμένων στους κάδους (γραμμές 9 έως 19). Ως εκ τούτου ο τρέχων κάδος bin γεμίζει με πόρους μέχρι να φτάσει το σημείο κορεσμού. Αρχικοποιείτε ένα σύνολο N_v με όλα τα στοιχεία τα οποία δεν έχουν ακόμη αποδοθεί σε κάποιον κάδο και τα οποία δεν παραβιάζουν τους περιορισμούς χωρητικότητας του τρέχοντα κάδου (γραμμή 10). Αν αυτό το σύνολο δεν είναι κενό τότε χρησιμοποιούμε έναν πιθανοτικό κανόνα επιλογής του επόμενου προς “πακετάρισμα” στοιχείου στον τρέχοντα κάδο (γραμμή 12).
4. Το στοιχείο έπειτα καταγράφεται στον πίνακα S_a , διαγράφεται από το σύνολο των στοιχείων IS και ενημερώνεται ο Host (γραμμές 13 έως 15). Αυτό εκτελείται μέχρις ότου υπάρχουν στοιχεία να αναθέσουμε στον τρέχοντα κάδο και τα οποία έχουν και αρκετή χωρητικότητα διαθέσιμη (γραμμή 9 και 10). Όταν όλα τα μυρμήγκια κατασκευάσουν τις λύσεις τους S_a γίνεται μια σύγκριση και επιλέγεται η καλύτερη (γραμμή 21). Η καλύτερη λύση γίνεται γνωστή σε όλο το σύστημα (γραμμές 22 έως 24) και βάσει αυτής ενημερώνονται τα μονοπάτια φερομόνης μεταξύ στοιχείων και κάδων (γραμμές 26 έως 34).
5. Βασική προϋπόθεση είναι ο σεβασμός ενός ανώτερου και ενός κατώτερου ορίου που έχει τεθεί. Για μονοπάτια φερομόνης μεταξύ στοιχείων- κάδων που έχουν πάρει μεγαλύτερη τιμή από το όριο, ξαναρχικοποιούνται στην max τιμή (γραμμές 28 έως 30). Ομοίως, όταν η τιμή της φερομόνης πέσει κάτω από το προκαθορισμένο όριο η τιμή min ενημερώνεται αναλόγως (γραμμές 31 έως 33). Ο αλγόριθμος τερματίζει μετά από nCycles και επιστρέφει την καλύτερη υπολογιστικά λύση S_{best} (γραμμή 36).

Algorithm 1 Energy-Aware ACO-based Workload Placement

```

1: Input: Set of items  $I$  and set of bins  $B$  with their associated resource demand vectors  $\vec{r}_i$  and  $\vec{C}_v$ 
   respectively, Set of parameters
2: Output: Global best solution  $S_{best}$ 
3:
4: Initialize parameters, Set pheromone value on all item-bin pairs to  $\tau_{max}$ 
5: for all  $q \in \{0 \dots nbCycles - 1\}$  do
6:   for all  $a \in \{0 \dots nbAnts - 1\}$  do
7:      $IS := I; v := 0$ 
8:      $S_a := [x_{i,j} := 0], \forall i \in \{0, \dots, m - 1\}, \forall j \in \{0, \dots, n - 1\}$ 
9:     while  $IS \neq \emptyset$  do
10:       $N_v := \{i \mid \sum_{j=0}^{n-1} x_{i,j} = 0 \wedge \vec{b}_v + \vec{r}_i \leq \vec{C}_v\}$ 
11:      if  $N_v \neq \emptyset$  then
12:        Choose item  $i \in N_v$  stochastically according to probability  $p_v^i := \frac{[\tau_{i,v}]^\alpha \times [\eta_{i,v}]^\beta}{\sum_{u \in N_v} [\tau_{u,v}]^\alpha \times [\eta_{u,v}]^\beta}$ 
13:         $x_{i,v} := 1$ 
14:         $IS := IS - \{i\}$ 
15:         $\vec{b}_v := \vec{b}_v + \vec{r}_i$ 
16:      else
17:         $v := v + 1$ 
18:      end if
19:    end while
20:  end for
21:  Compare ants solutions  $S_a$  according to the objective function  $f \rightarrow$  Save cycle best solution as  $S_{cycle}$ 
22:  if  $q = 0 \vee IsGlobalBest(S_{cycle})$  then
23:    Save cycle best solution as new global best  $S_{best}$ 
24:  end if
25:  Compute  $\tau_{min}$  and  $\tau_{max}$ 
26:  for all  $(i, B_v) \in I \times B$  do
27:     $\tau_{i,v} := (1 - \rho) \times \tau_{i,v} + \Delta_{i,v}^{best}$ 
28:    if  $\tau_{i,v} > \tau_{max}$  then
29:       $\tau_{i,v} := \tau_{max}$ 
30:    end if
31:    if  $\tau_{i,v} < \tau_{min}$  then
32:       $\tau_{i,v} := \tau_{min}$ 
33:    end if
34:  end for
35: end for
36: return Global best solution  $S_{best}$ 

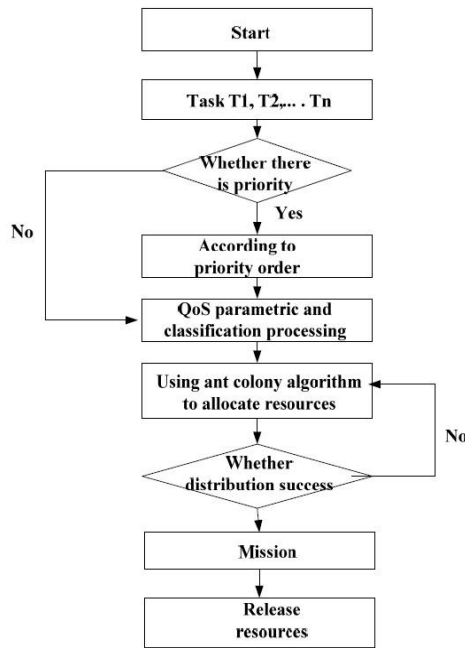
```

Εικόνα 5.3 Αλγόριθμος για φόρτο εργασίας

Τα πειράματα των ερευνητών έδειξαν ότι ο αλγόριθμος βασισμένος στην ευφυΐα του σμήνους των μυρμηγκιών παρέχει ανώτερα κέρδη στην ενέργεια σε σύγκριση με την παραδοσιακή άπληστη μέθοδο (greedy algorithm) τοποθέτησης φόρτου εργασίας.

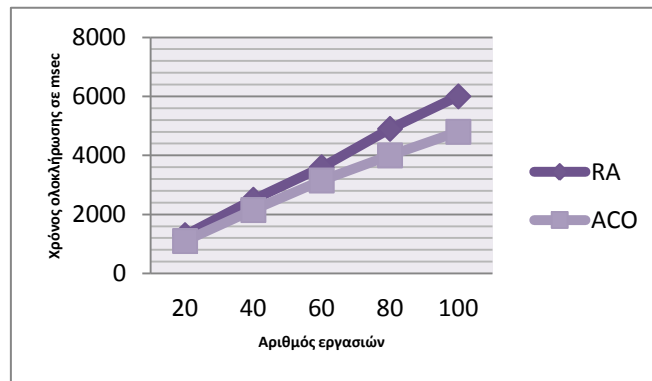
5.4.3 Στρατηγική χρονοδρομολόγησης βάση ACO για χρήση πόρων σε Clouds

Τον Σεπτέμβριο του 2012 οι Linan Zhu, Qingshui L και Lingna He παρουσίασαν μια μελέτη στρατηγικής χρονοδρομολόγησης πόρων σε Clouds η οποία ήταν εμπνευσμένη από τον αλγόριθμο ACO. Αρχικά, ανάλογα με τις απαιτήσεις του χρήστη, γίνεται μια ταξινόμηση των απαιτήσεων αυτών ανάλογα με τις προϋποθέσεις για QoS. Βάσει αυτής της ταξινόμησης και με την χρήση του αλγορίθμου ACO προγραμματίζεται πως θα γίνει η δέσμευση των πόρων. Μόλις οι αιτήσεις για πόρους πληρούν τις προϋποθέσεις για QoS και έχει βρεθεί το συντομότερο μονοπάτι τότε αρχίζει η δέσμευση και η εκτέλεση των εργασιών. Η διαδικασία παρουσιάζεται στο παρακάτω διάγραμμα ροής



Σχήμα 5.1 Διάγραμμα ροής

Βασισμένοι σε αυτή την στρατηγική εκτελέστηκαν διάφορα πειράματα. Τα πειράματα πραγματοποιήθηκαν πάνω στην πλατφόρμα CloudSim και έγινε σύγκριση μεταξύ του ACO και του τυχαία κατανομημένου αλγορίθμου (random distribution algorithm - RA)



Γράφημα 5.1 Σύγκριση ACO με RA

Από το σχήμα φαίνεται, πως όσο μεγαλύτερο είναι το μέγεθος των εργασιών τόσο λιγότερο χρόνο χρειάζεται ο ACO για να τις διεκπεραιώσει σε σύγκρισή με τον γενικό αλγόριθμο.

Κεφάλαιο 6ο

6 Συμπεράσματα - Επίλογος

Η ραγδαία έξαρση της τεχνολογίας με την ανάπτυξη των δικτύων και του ιντερνέτ οδήγησε τους ερευνητές στην κατασκευή αλγορίθμων προσπαθώντας να κατανοήσουν, να διαχειριστούν και να έχουν βέλτιστα οφέλη από αυτά. Τις τελευταίες δεκαετίες όλο και απομακρυνόμαστε από τα συμβατικά – σταθερού τύπου δίκτυα και προσανατολιζόμαστε στην ανάπτυξη αυτόνομων και αυτό οργανωμένων δικτύων όπως είναι τα ad-hoc και Clouds

Παρατηρώντας μια αποικία που αποτελείτε από μικρά, πολλά και αυτόνομα άτομα, προέκυψε ο αλγόριθμος ACO. Ο ACO έχει άπειρες εφαρμογές και ποικίλα οφέλη, από την κατασκευή αυτόνομων ρομποτικών εφαρμογών που μπορούν να χρησιμοποιηθούν σε στρατιωτικές εφαρμογές έως την διαχείριση υδάτινων και άλλων πόρων.

Η χρήση στα ad-hoc δίκτυα και Clouds αποτελεί σημαντικό κομμάτι έρευνας και πολλά πρωτόκολλα υλοποιήθηκαν βάσει αυτού. Τα οφέλη της χρήσης του σε δίκτυα ποικίλουν και χαρακτηριστικά αναφέρουμε, τον περιορισμό της ενέργειας που δαπανάτε σε ένα δίκτυο, αλλά και βέλτιστη τοποθέτηση του φόρτου εργασίας ενός δικτύου με ελάχιστο κόστος.

Βιβλιογραφία - Αναφορές

Βιβλιογραφία

- [1]. Ant colonies for the travelling salesman problem, Marco Dorigo , Luca Maria Gambardella, 24 October 1996
- [2]. Swarm intelligence, E. Bonabeau, M. Dorigo, G. Theraulaz, Oxford University Press, 1999.
- [3]. The Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization, McGraw-Hill, 11-32,1999
- [4]. Ant System: Optimization by a colony of cooperating agents. M. Dorigo, V. Maniezzo & A. Colorni, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41,1996.
- [5]. Elitist Ant System, M. Dorigo, Optimization, Learning and Natural Algorithms. Ph.D.Thesis, Politecnico di Milano, Italy, [in Italian], 1992.;
- [6]. Ant System: Optimization by a colony of cooperating agents M.Dorigo, V. Maniezzo and A. Colorni, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1), 29-41, 1996
- [7]. Ant Colony System M. Dorigo & L.M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1):53-66,1997. Max-Min Ant System T. Stützle and H. H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems. 16(8):889--914,2000.
- [8]. Ant algorithms and stigmergy, Marco Dorigo, Eric Bonabeaub, Guy Theraulaz , Future Generation Computer Systems 16 (2000) 851–871
- [9]. Ant colony optimization theory: A survey, Marco Dorigo, Christian Blum, 30 May 2005
- [10]. A review of ant algorithms, R.J. Mullen, D. Monekosso, S. Barman, P. Remagnino, Expert Systems with Applications 36 (2009)
- [11]. A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks, J. S. Baras and H. Mehta, Proc. WiOpt, Sophia-Antipolis, France, March 2003.
- [12]. GPS/Ant-like routing algorithm in ad hoc networks, Telecommunication Systems, D. Camara, A. Loureiro, 18, 1-3, 2001, 85-100.
- [13]. , Probabilistic guarantees and algorithms for ad hoc networks, manuscript, P. Eugster 2003.
- [14]. Ant-routing-algorithm (ARA) for mobile multi-hop ad-hoc networks - new features and results, Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks M. Günes, M. Kähler, and I. Bouazizi, (Med-Hoc-Net'2003), Mahdia, Tunisia, 25-27, June 2003.
- [15]. ARA - the ant colony based routing algorithm for manets, M. Gunes, U. Sorges, and I. Bouazizi, Proc. ICPP Workshop on Ad Hoc Networks IWAHN, 2002, 79-85.

- [16]. Ants-based routing in large scale mobile ad-hoc Networks, M. Heissenbüttel and T. Braun, Kommunikation in Verteilten Systemen KiVS03, Leipzig, Germany, February 25-28, 2003.
- [17]. A parallel ant colony optimization algorithm for all-pair routing in MANETs, M.T. Islam, P. Thulasiraman, R.K. Thulasiram, Proc. IEEE Int. Parallel and Distributed Processing Symposium IPDPS, Nice, France, April 2003.
- [18]. Accelerated ants routing in dynamic networks, H. Matsuo, K. Mori, 2nd Int. Conf. Software Engineering, Artificial Intelligence, Networking & Parallel Distributed Computing, Nagoya, Japan, 2001.
- [19]. A novel routing protocol using mobile agents and reactive route discovery for ad hoc wireless networks S. Marwaha, C. K. Tham, D. Srinivasan, Proc. IEEE ICON, 2002.
- [20]. Mobile agent based routing protocol for mobile ad hoc networks S. Marwaha, C. K. Tham, D. Srinivasan, Proc. IEEE GLOBECOM, 2002.
- [21]. S. Rajagopalan, C. Jaikaeo, and C.C. Shen, Unicast Routing for Mobile Ad hoc Networks with Swarm Intelligence, Technical Report #2003-07, University of Delaware, May 2003.
- [22]. Termite: Emergent ad-hoc networking, M. Roth and S. Wicker, Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003), Mahdia, Tunisia, 25-27, June 2003.
- [23]. Ant Routing System – a routing algorithm based on ant algorithms applied to a simulated network. Mikkel Bundgaard, Troels C. Damgaard, Federico Decara og Jacob W. Winther, Spring 2002 IT University of Copenhagen
- [24]. An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem, Zar Chi Su Su Hlaing, May Aye Khine, 2011 International Conference on Information Communication and Management
- [25]. Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm, Linan Zhu, Qingshui Li and Lingna He, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012
- [26]. Ant colony Optimization: A Solution of Load balancing in Cloud, Ratan Mishra and Anant Jaiswal, International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012
- [27]. Energy-Aware Ant Colony Based Workload Placement in Clouds, Eugen Feller, Louis Rilling, Christine Morin, "The 12th IEEE/ACM International Conference on Grid Computing (GRID-2011) (2011)"
- [28]. Simulation of Ant Routing Protocol for Ad-hoc networks in NS-2, FJ Arbona Bernat, November 2006
- [29]. Ant Colony Optimization for Routing in Mobile Ad Hoc Networks in Urban Environments, Gianni A. Di Caro, Frederick Ducatelle, and Luca M. Gambardella, Technical Report No. IDSIA-05-08, May 2008
- [30]. Ανακάλυψη πληροφορίας πλαισίου με χρήση νοημοσύνης σμήνους, Ε.Συνοδινός, Ι. Προβατάρης, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Αθήνα Οκτώβριος 2008
- [31]. Οικονομοτεχνικά μοντέλα για τη βέλτιστη διαχείριση απορριμμάτων, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και

Μηχανικών Υπολογιστών, Γ.Δευτεραίου, Μ. Κολοκάθη, Αθήνα, Σεπτέμβριος 2008

- [32]. Η μέθοδος βελτιστοποίησης με αποικίες μυρμηγκιών: Εφαρμογή σε διακριτά και συνεχή προβλήματα, Γ.Φαινέκου, Εθνικό Μετσόβιο Πολυτεχνείο, Τμήμα Μηχανολόγων Μηχανικών
- [33]. Γεωγραφικές Μέθοδοι Δρομολόγησης σε Ad Hoc Ασύρματα Δίκτυα, Άγγελος Φατούρος, Οικονομικό Πανεπιστήμιο Αθηνών, Ιούλιος 2009
- [34]. Μελέτη και ανάπτυξη αλγορίθμων εμπνευσμένων από την Βιολογία, Δ. Καρβούνας, Πανεπιστήμιο Πειραιώς, 2009
- [35]. Συστήματα Ανίχνευσης Επιθέσεων, Ανδρέας Κουρκοβέλης, Πανεπιστήμιο Πειραιώς, Φεβρουάριος 2011
- [36]. Αλγόριθμοι κατασκευής και μελέτη προβλημάτων χρωματισμού κλάσεων τέλειων γραφημάτων, Δήμητρα Τσάμη, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Φεβρουάριος 2010
- [37]. Μελέτη αποδοτικού μηχανισμού επικοινωνίας σε σμήνος που αποτελείται από διαμεσολαβητές περιορισμένων δυνατοτήτων, Π.Σταμάτης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, 2007
- [38]. Βελτιστοποίηση απόδοσης μηχανισμού διάσχισης του διαδικτύου (Web Crawler) με βάση το προβαλλόμενο περιεχόμενο στους χρήστες, Σ. Χαμπηλομάτης, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, 2012
- [39]. Μηχανισμός Ανακάλυψης Πληροφορίας Πλαισίου με χρήση Αλγορίθμων Διάχυσης Πληροφορίας, Μιχαήλ Σ. Νικητίδης, Ελληνικό Ανοικτό Πανεπιστήμιο, Πάτρα, απρίλιο 2009
- [40]. Robustness and Resilience in Communication Networks from Complex Systems Perspective, Π. Φωτίου, Πανεπιστήμιο Κύπρου Τμήμα Πληροφορικής
- [41]. Δυναμικές διεργασίες σε πολύπλοκα συστήματα με χρήση υπολογιστικών προσομοιώσεων, Αριστοτέλης Κίττας, Τμήμα Φυσικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Μάρτιος, 2011
- [42]. Serugendo - Engineering Self-Organising Systems (Springer, 2004)

Αναφορές – Πηγές από το internet

1. [Agent Based Models with Anylogic](#)
2. [Adaptive Routing taking Cues from Stigmergy in Ants « Onionesque Reality](#)
3. [MUTE: Simple, Anonymous File Sharing](#)
4. [Ant Colony Optimisation](#)
5. [A Java Framework for Ant Colony Systems - TechyLib](#)
6. [About ACO](#)
7. [ScienceDirect.com - Future Generation Computer Systems - Ant algorithms and stigmergy](#)
8. [Ant System - Clever Algorithms: Nature-Inspired Programming Recipes](#)
9. [Test Run - Ant Colony Optimization](#)
10. [Greedy Algorithms](#)
11. [Agent Computing and Multi-Agent Systems: 9th Pacific Rim International ... - Βιβλία Google](#)
12. [A trail of ants – MYRMECOS - Insect Photography - Insect Pictures](#)
13. [JavaACSFramework.pdf \(application/pdf Object\)](#)
14. [Ant-based routing in networks: AntNet and AntHocNet](#)
15. [Ant Colony Simulation](#)
16. [Simple ant routing algorithm strategies for a \(Multipurpose\) MANET model.pdf \(application/pdf Object\)](#)
17. [Travelling salesman problem - Wikipedia, the free encyclopedia](#)
18. [uchicago.src.sim.gui \(RepastJ API Documentation\)](#)
19. [Ant Colony Optimization: Routing](#)
20. [Ιστορική ανάλυση του Cloud Computing - Testwiki \(και επιπλέον αναφορές απο την συγκεκριμένη πηγή\)](#)
21. [Τα έξι επίπεδα υλοποίησης του Cloud - Testwiki](#)
22. [Ant colony optimization algorithms - Wikipedia, the free encyclopedia](#)
[Stigmergy - Wikipedia, the free encyclopedia](#)