ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΜΕΤΑ-ΕΥΡΕΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΙ ΕΦΑΡΜΟΓΗΣ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΣΥΝΔΥΑΣΤΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

ΒΛΑΧΟΣ ΑΡΙΣΤΕΙΔΗΣ

ΠΕΙΡΑΙΑΣ 2005

ΠΡΟΛΟΓΟΣ

Η βελτιστοποίηση αποτελεί ένα βασικό θέμα συζητήσεων στην πληροφορική, στην τεχνητή νοημοσύνη, στην επιχειρησιακή έρευνα, ενώ βρίσκει εκπληκτικές εφαρμογές στον τομέα της βιομηχανίας.

Τα προβλήματα που συναντώνται στον πραγματικό κόσμο, στη βιομηχανία και την επιστήμη είναι δύσκολα συνδυάστηκα προβλήματα βελτιστοποίηση (NP-hard combinatorial optimization problems). Η επίλυση αυτών των προβλημάτων με τη χρήση κλασσικών αιτιοκρατικών μεθόδων είναι, σχεδόν, αδύνατη. Γι' αυτό το λόγο η έρευνα κατευθύνθηκε προς τους στοχαστικούς ευρεστικούς αλγόριθμους που προξενούνται ή δημιουργούνται εμπνέονται από φυσικά, κοινωνικά και βιολογικά φαινόμενα.

Πρόσφατα, στα πλαίσια της παραπάνω έρευνας, αναπτύχθηκε μια αλγοριθμική πρακτική βασισμένη στη μέθοδο βελτιστοποίησης με αποικίες μυρμηγκιών (ant colony optimization / ACO) η οποία στηρίζεται στη συλλογική συμπεριφορά των μυρμηγκιών κατά την αναζήτηση και τη συγκομιδή της τροφής τους.

Η εφαρμογή των αλγορίθμων ACO σε διακριτά αλλά και σε συνεχή προβλήματα, με μεγάλη πρακτική αξία, είναι πολύ ενθαρρυντικά σε σύγκριση με αλγόριθμους πολύ μεγάλης «διάρκειας ζωής».

Οι στόχοι αυτής της έρευνας είναι :

1. Να εξερευνήσει και να παρουσιάσει τους μηχανισμούς που διέπουν μια κοινωνία εντόμων, και ειδικά την κοινωνία μυρμηγκιών. Στη συνέχεια να μελετηθεί η συμπεριφορά τους και η συμβολή τους στην ανάπτυξη συστημάτων τεχνικής νοημοσύνης μέσω ενός πανοράματος κυρίων αλγορίθμων.
2. Να δοθεί η μαθηματική θεμελίωση των μοντέλων διαχείρισης φερομόνης μέσω των εξισώσεων διαφορών, καθώς επίσης και εκείνης της εξάτμισης της φερομόνης μέσω γεννητριών συνάρτησης, εξετάζοντας την γεωμετρική και αλγεβρική της σύγκλισης.
3. Να δείξει τα αποτελέσματα της εφαρμογής των αλγορίθμων αποικίας μυρμηγκιών σε μια μεγάλη περιοχή προβλημάτων σημαντικής πρακτικής αξίας. Η αντιμετώπιση και ο τρόπος επίλυσης αυτών των προβλημάτων ανέδειξαν πρωτότυπους αλγορίθμους που συμβάλλουν στην καθολική αποδοχή της αποικίας μυρμηγκιών, σαν εργαλείο μελέτης, έρευνας και καθιέρωσης συγκεκριμένης μεθοδολογίας βελτιστοποίησης. Αυτή αφορά όχι μόνο πρόβλημα συνδυαστικής βελτιστοποίησης, αλλά, και συνεχή ή και συνδυασμό αυτών.

Τελειώνοντας θα ήθελα να ευχαριστήσω όλους τους αφανείς συντελεστές αυτής της διατριβής. Την γυναίκα μου, Κατερίνα, που μου πρόσφερε χρόνο, υπομονή και κατανόηση, προκειμένου να ολοκληρωθεί η προσπάθεια μου.

Το μεγαλύτερο ευχαριστώ, όμως, ανήκει στον επιβλέποντα καθηγητή μου, κ. Ευάγγελο Φούντα. Το βάθος της επιστημονικής του κατάρτισης και η αγάπη του και ο σεβασμός του για τη διδασκαλία της μαθηματικής επιστήμης ήταν για μένα μια πρωτόγνωρη εμπειρία. Ευχαριστώ επίσης τα δύο μέλη της τριμελούς επιτροπής: τον αντιπρύτανη κ. Δ. Δεσπότη και τον επίκουρο καθηγητή κ. Δ. Γκιζόπουλο.

Η εμπιστοσύνη που μου έδειξε, η καθοδήγηση που μου πρόσφερε και η υποστήριξη που μου παρέσχε, συνετέλεσαν καθοριστικά στην υλοποίηση αυτής της διατριβής.

Βλάχος Αριστείδης

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΚΕΦΑΛΑΙΟ 2

ΚΕΦΑΛΑΙΟ 3

ΚΕΦΑΛΑΙΟ 4

ΚΕΦΑΛΑΙΟ 5

ΚΕΦΑΛΑΙΟ 6

6 Μοντέλα Διαχείρισης Φερομόνης

ΚΕΦΑΛΑΙΟ 7

7 Μια διαφοροποιημένη μορφή του αλγορίθμου Ant Colony System (DFACS) για την επίλυση προβλημάτων δρομολόγησης οχημάτων (VRP)

ΚΕΦΑΛΑΙΟ 8

ΚΕΦΑΛΑΙΟ 9

ΚΕΦΑΛΑΙΟ 10

ΚΕΦΑΛΑΙΟ 11

11 Βέλτιστη λύση του προβλήματος γραμμικής τοποθέτησης μηχανών με τη χρήση αλγορίθμων αποικίας μυρμηγκιών

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Αιτιολόγηση

Η δυνατότητα μοντελοποίησης μερικών πραγματικών προβλημάτων, σαν διακριτά προβλήματα βελτιστοποίησης δημιούργησε την ανάγκη ανάπτυξης κατάλληλων εργαλείων ικανών να λύσουν αυτά τα προβλήματα, γρήγορα, και αποτελεσματικά. Τα διακριτά προβλήματα βελτιστοποίησης χαρακτηρίζονται από ένα υπολογίσιμο, πεπερασμένο, χώρο λύσεων (εφικτών λύσεων), όπου η τιμή μιας αντικειμενικής συνάρτησης συνδέεται με μια εφικτή λύση. Στα διακριτά προβλήματα βελτιστοποίησης, βρίσκονται λύσεις οι οποίες βασίζονται στις τιμές της αντικειμενικής συνάρτησης, ολοκληρωτικά μέσα στην εφικτή περιοχή του προβλήματος.

Μερικά διακριτά προβλήματα βελτιστοποίησης είναι δύσκολα (NP-hard), που σημαίνει ότι δεν υπάρχει αλγόριθμος που να τα λύνει σε πολυωνυμικό χρόνο. Υπάρχουν, όμως, τεχνικές βέλτιστων λύσεων, των διακριτών προβλημάτων βελτιστοποίησης. Σε μερικές περιπτώσεις, στην πράξη, είναι αναγκαία μια «καλή» λύση. Με άλλα λόγια, αναζητείται μια *υποβέλτιστη λύση* η οποία λαμβάνεται με την εφαρμογή *ευρεστικών* αλγορίθμων.

Τα τελευταία χρόνια, δημιουργήθηκε ένα νέο αλγοριθμικό τοπίο στο οποίο σημαντική θέση έχουν οι *μετα-ευρεστικοί* αλγόριθμοι, οι οποίοι λύνουν δύσκολα προβλήματα βελτιστοποίησης, με ικανοποιητικό τρόπο. Οι μετα-ευρεστικοί αλγόριθμοι περιλαμβάνουν τις τεχνικές:

- *Προσομοιωμένη Ανόπτηση*
  Simulated Annealing (SA)
  [Metropolis και άλλοι, 1953]
  [Kirkpatrick και άλλοι, 1983]
  [Cerny, 1985]

- *Γενετικοί Αλγόριθμοι*
  Genetic Algorithms (GA)
  [Holland, 1975]
  [Goldberg, 1989]

και

- *Tabu Search (TS)*
  [Glover, 1986]
  [Glover και Laguna, 1993]

Η επιτυχής εφαρμογή των τεχνικών που βασίζονται σε φυσικές οντότητες, όπως η προσομοιωμένη ανόπτηση, γενετικοί αλγόριθμοι και τα νευρωνικά δίκτυα, σε δύσκολα προβλήματα ενθάρρυναν πολλούς ερευνητές να τις ακολουθήσουν και να τις επεκτείνουν.

Πρόσφατα αναπτύχθηκαν τεχνικές, με συγκεκριμένες προτάσεις λύσεων των διαφόρων προβλημάτων βελτιστοποίησης που έχουν τις ρίζες τους στη φύση. Ένα υποσύνολο αυτών των τεχνικών αναπτύχθηκε στη βάση εννοιών που προέκυψαν από τα αποτελέσματα μελετών της συμπεριφοράς των κοινωνικών εντόμων.

Κοινωνικά, θεωρούνται τα έντομα που ζουν σε αποικίες, και η γενικότερη συμπεριφορά τους κατευθύνεται από την ανάγκη επιβίωσης της ομάδας και όχι του μεμονωμένου ατόμου. Η εκπληκτική πολύπλοκη συμπεριφορά των κοινωνιών τους και ο τρόπος λειτουργίας τους, έρχονται σε αντίθεση με την πολύ απλή δομή των ατόμων και τις περιορισμένες ικανότητες που διαθέτουν. Οπότε η κοινωνική ευφυΐα (Swarm intelligence) που αναπτύσσουν, για να λύσουν δύσκολα προβλήματα, αναδύεται από τον τρόπο με τον οποίο συνεργάζονται (emergent behavior).

Τα κοινωνικά άτομα εργάζονται χωρίς επίβλεψη. Η ομαδική τους εργασία, αυτο-οργανώνεται και ο συντονισμός αναδύεται από τις αλληλεπιδράσεις ανάμεσα στα άτομα της αποικίας. Οι αλληλεπιδράσεις αυτές, μπορεί να είναι πρωτόγονες, συνολικά, όμως, οι ενέργειες των ατόμων κατευθύνονται προς μία αποτελεσματική λύση για κάθε δύσκολο πρόβλημα. Η συλλογική συμπεριφορά που αναδεικνύεται από μια ομάδα κοινωνικών εντόμων ονομάζεται Swarm Intelligence.

Το πρόβλημα διανομής των υλικών αγαθών από μια αποθήκη ή ενός συνόλου αποθηκών σε πελάτες, γνωστό σαν πρόβλημα δρομολόγησης οχημάτων (VRP), είναι ένα δύσκολο πρόβλημα συνδυαστικής βελτιστοποίησης. Η διαχείριση ηλεκτρικού φορτίου στα Κέντρα Ελέγχου Ενέργειας καθώς και τα προβλήματα τοποθέτησης – κατανομής απαιτούν λύσεις, που στην παρούσα διατριβή, περνάνε μέσα από μετα-ευρεστικές τεχνικές που πηγάζουν από την κοινωνική ευφυΐα μιας αποικίας εντόμων.

1.2 Ανασκόπηση βιβλιογραφίας

Οι αλγόριθμοι που χρησιμοποιούν μερικές αναλογίες με τα φυσικά και κοινωνικά συστήματα παράγουν μη-αιτιοκρατικούς ευρεστικούς μεθόδους, ικανοί να προκαλούν «πολύ καλά» αποτελέσματα – λύσεις των δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης, ενώ δημιούργησαν ένα ερευνητικό πεδίο πολύ ενδιαφέρον.

Σε αυτή την παράγραφο θα γίνει μια σύντομη παρουσίαση αυτών των αλγορίθμων που συναντώνται στη διεθνή βιβλιογραφία. Αυτοί οι αλγόριθμοι εμπνέονται από φυσικά, κοινωνικά ή βιολογικά φαινόμενα.

1.2-1 Προσομοιωμένη Ανόπτηση – Simulated Annealing

Η Προσομοιωμένη Ανόπτηση (SA) [Metropolis και άλλοι, 1953] είναι ένας αλγόριθμος προσομοίωσης της ψύξης υλικού σε θερμό νερό, με αποτέλεσμα να προκύπτουν οι ελάχιστες δυνατές ενεργειακές καταστάσεις των μορίων του υλικού. Αυτή η διαδικασία ονομάζεται ανόπτηση (annealing). Κατά τη διάρκεια της διαδικασίας της ανόπτησης η θερμοκρασία μειώνεται αργά. Ο αλγόριθμος Metropolis προσομοιώνει την αλλαγή στην ενέργεια του υλικού, όταν υποβάλλεται σε μια διαδικασία ψύξης, μέχρι να συγκλίνει σε μια σταθερή παγωμένη κατάσταση.

Υπάρχει δυνατότητα σε κάθε θερμοκρασία να αλλάξει η ολική ενέργεια του υλικού όταν προκαλούνται μικρές μετατοπίσεις των μορίων του. Σε κάθε θερμοκρασία, αυτή η διαδικασία ονομάζεται *διαταραχή*. Για κάθε θερμοκρασία απαιτείται ένας μεγάλος αριθμός διαταραχών, κάθε μία από αυτές δίνει διάφορα ολικά ενεργειακά επίπεδα. Πρόβλημα: ποια διαταραχή θα δώσει το χαμηλότερο ενεργειακό επίπεδο σε μια θερμοκρασία;

Η μεθοδολογία που περιγράφεται, παρακάτω, διαμορφώνει την απάντηση στο πρόβλημα. Μετά ην διαταραχή, υπολογίζεται το μέτρο της ενεργειακής μεταβολής (ΔΕ). Στην περίπτωση που $\Delta E < 0$, η νέα θέση των μορίων χαρακτηρίζεται από το χαμηλότερο ενεργειακό επίπεδο, ενώ όταν $\Delta E > 0$, η νέα θέση των μορίων χαρακτηρίζεται από το μεγαλύτερο ενεργειακό επίπεδο. Αυτό εκφράζεται μέσω μιας πιθανότητας (η πιθανότητα αυξάνεται με τη θερμοκρασία) όπου μετά την διαταραχή της διάταξης αυτών των μορίων προκύπτει μείωση της ενέργειας. Οι νόμοι της θερμοδυναμικής δηλώνουν ότι στη θερμοκρασία Τ η πιθανότητα αύξησης του μέτρου της ενέργειας ΔΕ δίνεται:

$$P(\Delta E) = e^{-\frac{\Delta E}{KT}} \qquad (1.1)$$

όπου:

ΔΕ είναι το μέτρο μεταβολής της ενέργειας
Κ μία σταθερά γνωστή ως σταθερά Boltzman
Τ είναι η τρέχουσα θερμοκρασία

Σε κάθε θερμοκρασία είναι δυνατό να προκύπτει ένας μεγάλος αριθμός διαταραχών, αλλά μετά από έναν αριθμό διαταραχών, το μέτρο μεταβολής της συνολικής ενέργειας είναι πολύ μικρό.

Σε αυτό το σημείο το υλικό αποκτά *θερμική ισορροπία* γι' αυτή τη σταθερή θερμοκρασία. Το επόμενο βήμα είναι να μειωθεί η θερμοκρασία και να δημιουργούνται διαταραχές ακόμα με σκοπό να αυξηθεί η θερμική ισορροπία στη συγκεκριμένη θερμοκρασία. Η διαδικασία τελειώνει όταν η διαδικασία αναζήτησης της χαμηλότερης θερμοκρασίας εξαντλείται.

Πριν εφαρμοστεί αυτός ο αλγόριθμος πρέπει να οριστούν οι παρακάτω τιμές:

- ο συνολικός αριθμός θερμοκρασιών
- η αρχική τιμή θερμοκρασίας
- η ποσότητα μεταβολής της ενέργειας που λαμβάνεται σαν η πιο χαμηλή. Είναι ανάγκη να καθοριστεί πότε απαιτείται η θερμική ισορροπία.

Ο αλγόριθμος Metropolis, μπορεί να εφαρμοστεί στα προβλήματα βελτιστοποίησης με αντικατάσταση των στοιχείων της φυσικής διαδικασίας ψύξης με τα στοιχεία ενός συνδυαστικού προβλήματος βελτιστοποίησης:

Θερμοδυναμική προσομοίωση	*Συνδυαστική βελτιστοποίηση*
Καταστάσεις συστήματος	Εφικτές λύσεις
Ενέργεια	Αντικειμενική συνάρτηση
Κατάσταση μετάβασης	Γειτονική λύση
Θερμοκρασία	Παράμετρος ελέγχου
Σημείο ψύξης	Ευρεστική λύση

Ο ψευδοκώδικας [Eglese, 1990] του αλγορίθμου (SA) είναι:

Select an initial state $i \in S_j$

Select an initial temperature $T > O_j$

Set temperature change counter $t := O_j$

Repeat

Set repetition counter $n : O_j$

Repeat

Generate state j,a neighbor of i;

Calculate $\delta := f(j) - f(i)$

if $\delta < 0$ then $i := j$

else if random $(0,1) < \exp(-\delta/T)$ then $i : -j$;

$\mathrm{Inc}(n)$;

Until $n = N(t)$;

$\mathrm{Inc}(t)$;

$T := T(t)$;

Until stopping criterion true.

όπου:

S	είναι ένα πεπερασμένο σύνολο τιμών,
i	είναι η προηγούμενη λύση,
j	είναι η επόμενη λύση,
$f(x)$	είναι η τιμή – κριτήριο για τη λύση x, και
$N(t)$	είναι ο αριθμός διαταραχών στην ίδια θερμοκρασία

1.2-2 Εξελικτικοί αλγόριθμοι (ΕΑ)

Οι εξελικτικοί αλγόριθμοι (ΕΑ) περιλαμβάνουν ένα σύνολο τεχνικών οι οποίες χρησιμοποιούνται για τη λύση δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης. Οι λειτουργίες τους εμπνέονται από φυσικές εξελικτικές διαδικασίες.

Ο ψευδοκώδικας ενός (ΕΑ) [Hertz και Kobler, 2000] είναι:

Generate an initial population of individuals;

While no stopping condition is met do
    Co-operation;
    Sell-adaptation;
End While

Στην οικογένεια των εξελικτικών Αλγορίθμων περιλαμβάνονται: οι Γενετικοί Αλγόριθμοι (EA), Scatter Search, Ant Systems και Adaptive Memory Algorithms.

1.2-2.1 Γενετικοί Αλγόριθμοι (EA)

Οι Γενετικοί Αλγόριθμοι [Holland, 1975] εμπνέονται από βιολογικές διαδικασίες. Βασίζονται στην ύπαρξη ενός πληθυσμού ατόμων, τα οποία αποτελούν τις λύσεις ενός προβλήματος. Οι Γενετικοί Αλγόριθμοι λειτουργούν με βάση τρεις τελεστές.

- Τελεστής *επιλογής* (selection) ο οποίος καθορίζει πια άτομα (λύσεις του προβλήματος βελτιστοποίησης) θα επιζήσουν στον πληθυσμό (σύνολο λύσεων) και θα διαδίδονται στους μελλοντικούς πληθυσμούς.

- Τελεστής *διασταύρωσης* (crossover) ο οποίος συνδυάζει δύο άτομα έτσι ώστε να προκύψουν δύο διαφορετικά άτομα, τα οποία έχουν μεγάλη πιθανότητα να έχουν καλύτερη καταλληλότητα από τους γονείς τους.

- Τελεστής *μετάλλαξης* ο οποίος επιτρέπει να ελεγχθούν νέες περιοχές της επιφάνειας ελέγχου (επιφάνεια ανίχνευσης της βέλτιστης λύσης).

Ο ψευδοκώδικας ενός Γενετικού Αλγορίθμου [Hertz και Kobler, 2000] είναι:

Chose an even integer $p \geq 2$ and generate an initial population P_0 of p individuals;
Set $i := 0$; (iteration counter);
While no stopping criteria is met do
    $\mathrm{Inc}(i)$ and initialize P_i to the empty set;
    While P_i has less than p individuals do
        Select two individuals I_1 and I_2 in P_{i-1};

Apply the crossover operator to I_1 and I_2 for creating offspring O_1 and O_2;
Add O_1 and O_2 to P_i;
End While
Apply the mutation operator to each individual in P_i;
End While

Ο τελεστής *επιλογής* ενεργοποιεί τον μηχανισμό εκμετάλλευσης (exploitation) του χώρου των λύσεων όπου πραγματοποιείται τοπική αναζήτηση βέλτιστων λύσεων σε περιοχές που έχουν ήδη δώσει καλά αποτελέσματα. Οι υπόλοιποι τελεστές ενεργοποιούν το μηχανισμό εξερεύνησης (exploration) όλου του δυνατού χώρου των λύσεων για την αποτελεσματικότερη αναζήτηση νέων κατάλληλων περιοχών.

1.2-2.2 Scatter Search

Ο αλγόριθμος Scatter Search [Glover, 1994] είναι μία στρατηγική αναζήτησης η οποία βασίζεται σε δύο σύνολα σημείων σε μία περιοχή λύσης. Το πρώτο είναι ένα σύνολο *σημείων αναφοράς* από το οποίο η επανάληψη θα δημιουργήσει ένα σύνολο *σημείων διασποράς* τα οποία μπορούν να λειτουργήσουν σαν σημεία αναφοράς μαζί με μερικά από τα προηγούμενα σημεία αναφοράς για την επόμενη επανάληψη.

Κάθε σημείο του συνόλου των σημείων διασποράς ορίζεται μέσω καθορισμένων βημάτων:

1. Κάνε ένα γραμμικό συνδυασμό του υποσυνόλου των τρεχόντων σημείων αναφοράς (έτσι ορίζεται το δοκιμαστικό σημείο – trial point)

2. Εφάρμοσε μια διορθωτική διαδικασία αν είναι αναγκαίο (το δοκιμαστικό σημείο μπορεί να είναι εκτός της εφικτής – ορισμένης – περιοχής και η διαδικασία επιδιόρθωσης πρέπει να εντάξει αυτό το σημείο στην εφικτή περιοχή).

3. Εφάρμοσε έναν βελτιωμένο αλγόριθμο ώστε να προκύψει ένα σημείο καλύτερης ποιότητας.

Ο ψευδοκώδικας για τον αλγόριθμο Scatter Search [Hertz και Kobler,

2000] είναι:

Generate an initial set R_0 of reference points
Set $i := 0$
While no stopping criteria is met do $\mathrm{Inc}(i)$;
Determine a Set T_i of trial points by making linear combinations of points in R_{i-1};
Transforms the trial points in T_i into set set f_i in order to obtain a set D_i of dispersed points;
Select points in $R_{i-1} \cup D_i$ to be put in the new set R_i of reference points;
End While.

1.2-2.3 Ant Systems (ASs)

Οι αλγόριθμοι ASs [Dorigo και άλλοι, 1996] εμπνέονται από την πραγματική αποικία μυρμηγκιών. Αυτοί οι αλγόριθμοι περιλαμβάνουν τεχνητά μυρμήγκια τα οποία διαφέρουν από τα φυσικά:

- τα τεχνητά μυρμήγκια έχουν μνήμη
- τα τεχνητά μυρμήγκια ζουν σε περιβάλλον όπου ο χρόνος είναι διακριτό μέγεθος

Οι παραπάνω αλγόριθμοι βασίζονται στην έμμεση επικοινωνία των μυρμηγκιών μέσω μιας χημικής ουσίας – φερομόνη – κατά την αναζήτηση τροφής. Στα παρακάτω κεφάλαια θα γίνει μελέτη, σε βάθος, αυτών των αλγορίθμων.

Γενικά, οι αλγόριθμοι που ανήκουν στην αποικία μυρμηγκιών αποτελούν την πλέον επιτυχημένη αντιμετώπιση και λύση των δύσκολων προβλημάτων.

1.2-2.4 Adaptive Memory Algorithms

Οι αλγόριθμοι Adaptive Memory [Rochat και Trailard, 1995] βασίζονται στην τεχνική του αλγορίθμου Tabu Search. Αυτή η τεχνική λειτουργεί με κεντρική μνήμη που συγκεντρώνει τις καλύτερες συνιστώσες των λύσεων που βρέθηκαν στη διάρκεια της αναζήτησης. Αυτές οι συνιστώσες συνδυάζονται για να δημιουργηθεί μια νέα λύση. Όταν η νέα λύση δεν είναι εφικτή, αλλάζει χρησιμοποιώντας μια διαδικασία επιδιόρθωσης. Σε κάθε νέα εφικτή λύση, κάποια μέθοδος βελτίωσης όπως ο αλγόριθμος TS ή κάποια άλλη τεχνική εφαρμόζεται. Τότε κάποιες συνιστώσες από όλες τις νέες λύσεις επιλέγονται να γίνουν οι υποψήφιες που θα αντικαταστήσουν τις παλιές συνιστώσες στην κεντρική μνήμη.

Ο ψευδοκώδικας του αλγορίθμου Adaptive Memory [Hertz και Kobler, 2000] είναι:

Generate a set of solutions and introduce their components in the central memory;
While no stopping criteria is met do
    Combine components of the central memory in order to create new solutions;
    Apply an improvement algorithm on each new solution;
    Update the central memory by removing some components and introducing new ones originating from the new feasible solutions;
End While.

1.2-3 Tabu Search (TS)

TS [Glover, 1993] είναι ένας αλγόριθμος τοπικής έρευνας και χρησιμοποιείται για τη λύση προβλημάτων της μορφής:

$$\min f(x)$$
$$x \in X$$

Η βασική ιδέα προέρχεται από μια αρχική λύση $x_0 \in X$ – η οποία προκύπτει με οποιοδήποτε τρόπο – και φθάνει σε μια λύση αρκετά καλή μέσω ενός αριθμού επαναλήψεων. Μεταξύ μιας επανάληψης και της άλλης, δύο λύσεις συνήθως υπάρχουν: η τρέχουσα και η καλύτερη ευρεθείσα.

Αν η τρέχουσα λύση στην αρχή της επανάληψης i είναι x_i, τότε είναι

δυνατό να βρεθεί μια νέα λύση που θα ανήκει στο X με μία διαφοροποίηση του (λύση) x_i. Αυτή η λύση είναι η λύση στη γειτονιά του x_i. Έστω ότι $N(x_i)$ το σύνολο όλων των λύσεων στη γειτονιά της λύσης x_i [Panta Lŭcic, 1999].

Γενικά, συμβαίνει σε όλους τους αλγορίθμους τοπικής αναζήτησης, αν, κατά τη διάρκεια της εξερεύνησης της γειτονιάς $N(x_i)$ μιας λύσης x_i, ανακαλύπτεται μια λύση καλύτερη από την τρέχουσα βέλτιστη, γίνεται εξερεύνηση στη γειτονιά της καλύτερης λύσης.

Αν η τρέχουσα λύση είναι τοπικό ελάχιστο, ο αλγόριθμος τοπικής αναζήτησης θα σταματήσει. Για να ξεπεραστούν τα τοπικά ελάχιστα, είναι αναγκαίο να κρατηθούν λύσεις που θα προκαλέσουν μια χειροτέρευση της τρέχουσας λύσης, απομακρυνόμενοι αρκετά μακριά από το τοπικό ελάχιστο.

Υποθέστε ότι βρισκόμαστε σε ένα τοπικό ελάχιστο x_i, πρέπει να επιλέξουμε σε ποια λύση πρέπει να μετατοπιστούμε, από τη στιγμή που καμία, στο σύνολο $N(x_i)$, δεν θα είναι καλύτερη από την x_i.

Στους αλγορίθμους TS επιλέγεται η μετατόπιση σε μια λύση $x_{i+1} \in N(x_i)$ στην οποία αντιστοιχεί μια ελάχιστη χειροτέρευση της αντικειμενικής συνάρτησης. Σε αυτό το σημείο δημιουργείται το πρόβλημα που βασίζεται η ιδέα του αλγορίθμου TS. Αν μετατοπιστούμε στη λύση x_{i+1}, και εξερευνούμε την γειτονιά της, είναι πολύ πιθανό η καλύτερη λύση στο σύνολο $N(x_{i+1})$ να είναι η x_i, δηλαδή από αυτή που πρέπει να απομακρυνθούμε. Όταν επιτρέπουμε λύσεις που δεν καλυτερεύουν την αντικειμενική συνάρτηση, δημιουργείται κίνδυνος, έστω και αν έχει γίνει ένας αριθμός κινήσεων να επιστρέψουμε σε λύση που έχουμε, ήδη, επισκεφτεί. Έτσι προκύπτει η ανάγκη χρήσης της μνήμης. Διατηρώντας στη μνήμη τις τελευταίες λύσεις από τις οποίες προέκυψε η τωρινή λύση, απαγορεύεται, για ένα ορισμένο χρονικό διάστημα, σε αυτές τις λύσεις να καταλαμβάνουν σημεία που ήδη έχουν επισκεφτεί.

Ενώ στην τοπική αναζήτηση αναφερόμαστε στη γειτονιά $N(x_i)$ μιας λύσης x_i, στους αλγορίθμους TS αποκλείονται, από την εξερεύνηση του συνόλου $N(x_i)$, μερικές λύσεις, ανάλογα με τον αριθμό επαναλήψεων που οδηγεί στη λύση x_i. Οπότε, το σύνολο των λύσεων που πρέπει να εξερευνηθούν, εξαρτάται, εκτός από τη λύση x_i, από την επανάληψη i της

αντίστοιχης μεθόδου, δηλαδή η γειτονιά είναι $N(x_i, i)$.

Για την υλοποίηση αυτής της έννοιας, είναι η αναγκαία μιας Tabu list. Η Tabu list είναι μια λίστα στην οποία απομνημονεύονται οι τελευταίες λύσεις. Ενώ, αποκλείονται οι λύσεις εκείνες που οδηγούν σε σημεία που έχουν ή επισκεφτεί.

Τελικά στον αλγόριθμο TS εισάγεται μια μνήμη που χρησιμοποιείται για την απομνημόνευση των προηγούμενων βημάτων που ενεργοποιούν τη διαδικασία βελτιστοποίησης και ιδιαίτερα αυτών που βελτίωσαν τη λύση του προβλήματος.

Ο ψευδοκώδικας [Dammeyer και Vob, 1993] του αλγορίθμου TS είναι:

Set iteration counter $i := 1$
Select an initial state $x_i \in X$ and initial objective value $f(x_i)$
Let $x^* := x_i$ and $f^* := f(x_i)$
While stopping criterion is not fulfilled do
    Select best admissible move (based to tabu list) that transforms x_i into x_{i+1} with objective function value $f(x_{i+1})$;
    Perform tabu list management;
    Update tabu list.
    Replace Solution x_i with x_{i+1} and objective value $f(x_i)$ with $f(x_{i+1})$;
    If $f(x_{i+1}) > f^*$ then $x^* := x_{i+1}$ and $f^* := f(x_{i+1})$;
    $Inc(i)$
End While

1.2-4 Νευρωνικά Δίκτυα

Τα τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Nets) [Hopfield και Tank, 1985] εμπνέονται από τη δομή και λειτουργία του ανθρώπινου εγκεφάλου.

Ο ανθρώπινος εγκέφαλος αποτελείται, περίπου, από 10^{11} νευρώνες

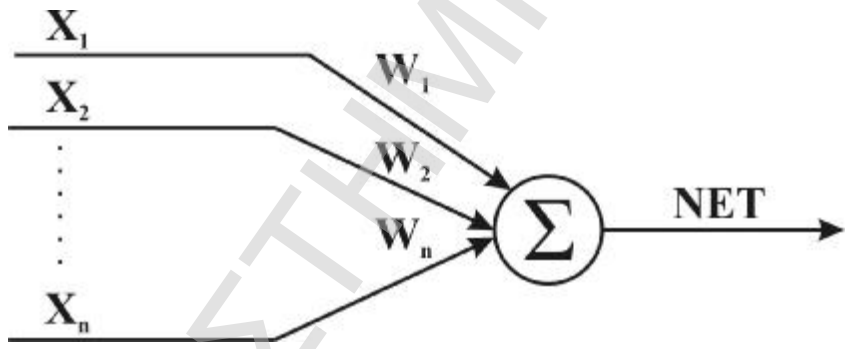
όπου ο καθένας έχει την ικανότητα να δέχεται και να εκπέμπει ηλεκτροχημικά σήματα.

Η βασική μονάδα ενός ANN είναι ο τεχνητός νευρώνας ο οποίος προσομοιώνει ιδιότητες ενός βιολογικού νευρώνα.

Οι τεχνητοί νευρώνες δέχονται έναν ορισμένο αριθμό σημάτων στην είσοδό τους ενώ εκπέμπουν έναν, ορισμένο, αριθμό σημάτων στην έξοδό τους. Αν $x_1, x_2, \mathbf{K}, x_n$ τα σήματα εισόδου ενός τεχνητού νευρώνα, σχ. 1.1, τότε το σήμα εξόδου (NET) θα δίνεται:

$$\mathrm{NET} = w_1 x_1 + w_2 x_2 + \mathbf{L} + w_n x_n$$

όπου w_i είναι ένας συντελεστής βάρους που εκφράζει τη σχέση σύνδεσης μεταξύ του τεχνητού νευρώνα i και του τρέχοντα νευρώνα.



Σχήμα 1.1: Δομή τεχνητού νευρώνα

Ο πρώτος τεχνητός νευρώνας προτάθηκε από τον McCulloch and Pitts (1943). Κάθε νευρώνας εκπέμπει μόνον ένα σήμα σε κάθε χρονική στιγμή, αλλά προς την κατεύθυνση πολλών νευρώνων.

1.3 Στόχοι της έρευνας

Με βάση τη βιβλιογραφική ανασκόπηση που προηγήθηκε, προκύπτει ότι ο καλύτερος τρόπος αντιμετώπισης των δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης στηρίζεται στη συλλογική συμπεριφορά που επιδεικνύεται από μια ομάδα κοινωνικών εντόμων η οποία αποκαλείται *κοινωνική ευφυΐα* (swarm intelligence). Άλλωστε αυτή αναδύεται από τον τρόπο με τον οποίο συνεργάζονται (emergent behavior) τα κοινωνικά έντομα, προκειμένου να επιλύσουν δύσκολα προβλήματα.

Οι αποικίες των κοινωνικών εντόμων και ειδικά της αποικίας μυρμηγκιών, αποτελούν επιτυχημένα παραδείγματα μιας ευρύτερης οικογένειας όπου εμφανίζονται οι συμπεριφορές που συναντώνται στα πολύπλοκα συστήματα (complex systems). Πολύπλοκα συστήματα είναι το διαδίκτυο, οι αγορές προϊόντων, τα συγκοινωνιακά δίκτυα κ.ά. Το όνομά τους το οφείλουν στην πολυπλοκότητα της συμπεριφοράς τους, που προκαλείται από το μεγάλο πλήθος διακριτών οντοτήτων στις οποίες συνίστανται.

Όπως και στα πολύπλοκα συστήματα, οι αποικίες εντόμων αποτελούνται από απλά συστατικά μέλη. Τα μέλη των αποικιών λειτουργούν κάτω από την έλλειψη πληροφόρησης για το όλο σύστημα, διαθέτουν μόνο, χωρικά, τοπική γνώση που αποκτούν με την αλληλεπίδρασή τους με τα άλλα μέλη ή το περιβάλλον. Βελτιώνουν την απόδοσή τους εμφανίζοντας εξειδικευμένες συμπεριφορές, λειτουργώντας και δρώντας σε παραλληλία. Αποτέλεσμα το συνολικό έργο που παράγεται από την αποικία, είναι σημαντικότερο και πέρα από το άθροισμα του έργου που παράγουν τα μέλη της (emergent behavior).

Τα μέλη μιας αποικίας λειτουργούν σαν αυτόνομοι πράκτορες (autonomous agents). Δεν εκτελούν εντολές από κάποιο αρχηγό, ούτε γνωρίζουν ένα καθολικό σχέδιο το οποίο καλούνται να εκτελέσουν. Η συντονισμένη δράση των οποίων χάρη στις τοπικές και απλές αλληλεπιδράσεις, οδηγεί σε καθολικά οργανωμένα συστήματα με βασικές αρχές αυτο-οργάνωσης.

Ο πρώτος στόχος της διατριβής είναι η εξερεύνηση, και η παρουσίαση των μηχανισμών που διέπουν μια κοινωνία εντόμων, και ειδικά της κοινωνίας μυρμηγκιών. Στη συνέχεια μελετάται η συμπεριφορά τους και η συμβολή τους στην ανάπτυξη συστημάτων τεχνικής νοημοσύνης μέσω ενός πανοράματος κυρίων αλγορίθμων.

Ένα από τα πιο σημαντικά αποτελέσματα των αναπτυσσόμενων

τεχνητών συστημάτων (Artificial Systems) που βασίζεται στην κοινωνική ευφυΐα (Swarm Intelligence), είναι η ανάπτυξη των αλγορίθμων αποικίας μυρμηγκιών. Τα τεχνητά μυρμήγκια [Dorigo και άλλοι, 1996] αναζητούν λύσεις, εμπνεόμενα από τη συμπεριφορά των πραγματικών μυρμηγκιών όταν εγκαταλείπουν τη φωλιά τους, για να αναζητήσουν την τροφή τους. Ακολουθούν την πιο σύντομη διαδρομή. Ο μηχανισμός, με βάση τον οποίο διαμορφώνεται η τάση των μυρμηγκιών να επιλέγουν τη σύντομη διαδρομή λειτουργεί με τη χρήση φερομονών. Κάθε μυρμήγκι, στην πορεία του, εναποθέτει ποσότητες φερομόνης, η οσμή της οποίας του επιτρέπει να βρει το δρόμο της επιστροφής στη φωλιά του αλλά και για να επικοινωνήσει με τα υπόλοιπα μυρμήγκια «υποδεικνύοντας» την πιθανή νέα πηγή τροφής.

Κάθε μυρμήγκι που έλκεται από την οσμή της φερομόνης ενισχύει την οσμή της εναποθέτοντας και αυτό ποσότητα φερομόνης, οπότε η διαδρομή γίνεται όλο και πιο αισθητή και στα άλλα μυρμήγκια. Με αυτό τον τρόπο τα μυρμήγκια ακολουθούν την κυρίαρχη διαδρομή η οποία είναι η πιο σύντομη. Με τον ίδιο τρόπο συμπεριφέρονται και τα τεχνητά μυρμήγκια. Παράλληλα, ένας επιπλέον μηχανισμός επιτρέπει στην αποικία της κοινότητας να «ξεχνά» διαδρομές που δεν χρησιμοποιούνται είτε γιατί εξαντλήθηκε η πηγή τροφής είτε γιατί έχει εντοπιστεί η σύντομη διαδρομή. Στη συνέχεια ενεργοποιείται ο μηχανισμός της εξάτμισης προκειμένου να εξατμίσει, στις μη χρήσιμες διαδρομές, τη φερομόνη που έχει εναποτεθεί από τα μυρμήγκια. Η εξάτμιση της φερομόνης αποκτά ιδιαίτερο ενδιαφέρον στη λειτουργία της αποικίας των μυρμηγκιών γιατί δίνει τη δυνατότητα στα μυρμήγκια να ξεχνούν τις λανθασμένες επιλογές τους και να προσαρμόζονται σε ένα γεωγραφικά και χρονικά μεταβαλλόμενο περιβάλλον. Τελικά, η αποτελεσματική απόδοση της αποικίας μυρμηγκιών, στη λύση διακριτών προβλημάτων συνδυαστικής βελτιστοποίησης, βασίζεται στη διαχείριση της φερομόνης.

Ο δεύτερος στόχος της διατριβής είναι η μαθηματική θεμελίωση μοντέλων διαχείρισης φερομόνης μέσω των εξισώσεων διαφορών και στη συνέχεια η μαθηματική θεμελίωση του μοντέλου της εξάτμισης της φερομόνης μέσω γεννητριών συνάρτησης, εξετάζοντας τη γεωμετρική και αλγεβρική της σύγκλιση.

Ενδιαφέρουσα αλλά και αποτελεσματική είναι η εφαρμογή των αλγορίθμων αποικίας μυρμηγκιών σε πολύπλοκα συστήματα. Η κλάση αυτών των συστημάτων παρουσιάζει σημαντικές ιδιότητες που εμφανίζονται στην περίπτωση αναζήτησης εγγενούς αποκεντρωμένης λύσης των προβλημάτων.

Ο τρίτος στόχος της διατριβής είναι η εφαρμογή αλγορίθμων αποικίας

μυρμηγκιών σε προβλήματα μεγάλης πρακτικής αξίας. Η αντιμετώπιση και ο τρόπος επίλυσης των προβλημάτων ανέδειξαν πρωτότυπους αλγορίθμους που συμβάλλον στην καθολική αποδοχή της αποικίας μυρμηγκιών, σαν εργαλείο μελέτης, έρευνας και καθιέρωσης συγκεκριμένης μεθοδολογίας βελτιστοποίησης. Αυτή αφορά όχι μόνον διακριτά προβλήματα συνδυαστικής βελτιστοποίησης αλλά, και συνεχή ή και συνδυασμό αυτών.

1.4 Δομή Κεφαλαίων της διατριβής

Η παρούσα διατριβή αποτελεί μια συμβολή:

i) στο πεδίο της εξερεύνησης των μηχανισμών της αποικίας μυρμηγκιών,

ii) στη μαθηματική θεμελίωση, του σημαντικότερου στοιχείου πληροφόρησης της αποικίας μυρμηγκιών, της φερομόνης,

και

iii) στη λύση προβλημάτων που αφορούν αξιόλογα πρακτικά προβλήματα.

Ειδικότερα προτείνονται μαθηματικά μοντέλα διαχείρισης φερομόνης μέσω εξισώσεων διαφορών, ενώ θεμελιώνεται το μαθηματικό μοντέλο της εξάτμισης της φερομόνης μέσω γεννήτριας συνάρτησης, εξετάζοντας την αλγεβρική και γεωμετρική της σύγκλιση.

Μελετήθηκαν και λύθηκαν, το

1. πρόβλημα δρομολόγησης οχημάτων,

2. πρόβλημα διαχείρισης ηλεκτρικού φορτίου στα Κέντρα Ελέγχου Ενέργειας,

και

3. πρόβλημα χωροθέτησης – κατανομής.

Η στρατηγική λύσης των παραπάνω προβλημάτων ανέδειξε πρωτότυπους αλγορίθμους που εμπλουτίζουν την εμπειρία της αποικίας μυρμηγκιών. Αναλυτικότερα:

Στο *Κεφάλαιο 1* αιτιολογώ την ανάγκη λύσης δύσκολων

προβλημάτων μέσω αποικίας κοινωνικών εντόμων. Γίνεται μια σύντομη βιβλιογραφική ανασκόπηση των κυρίων μετα-ευρεστικών αλγορίθμων και μεθοδολογιών. Στη συνέχεια αναλύονται οι στόχοι της διατριβής.

Στο *Κεφάλαιο 2* αναφέρομαι στη φύση των προβλημάτων συνδυαστικής βελτιστοποίησης, υπογραμμίζοντας κάποιες κατηγορίες τους. Στη συνέχεια δίνεται έμφαση στις στοχαστικές επαναληπτικές μεθόδους σαν κύρια εργαλεία λύσης των προβλημάτων βελτιστοποίησης, με ιδιαίτερη επισήμανση στη μεγάλη αξία των μετα-ευρεστικών αλγορίθμων.

Στο *Κεφάλαιο 3* περιγράφονται οι μηχανισμοί διεκπεραίωσης συλλογικών εργασιών και μάλιστα με βέλτιστο τρόπο που βρίσκονται στην αυτο-οργάνωση των μυρμηγκιών, ενώ μελετάται η διαδικασία έμμεσης επικοινωνίας των μυρμηγκιών όπως ορίζεται από τον όρο *στιγμεργία*. Τέλος προτείνεται ένα στοχαστικό μοντέλο, μέσω εξισώσεων διαφορών, εναπόθεσης φερομόνης στο κλασικό πείραμα της ασύμμετρης γέφυρας.

Στο *Κεφάλαιο 4* περιγράφονται οι κλασικοί αλγόριθμοι αποικίας μυρμηγκιών που εφαρμόζονται στο πρόβλημα του περιοδεύοντος πωλητή (TSP): Ant System (AS), Ant Colony System (ACS) και Max-Min Ant System (MMAS). Στην περίπτωση του αλγορίθμου MMAS προτείνεται μια ενδιαφέρουσα απόδειξη του κάτω ορίου φερομόνης $(\tau_{\min})$.

Στο *Κεφάλαιο 5* περιγράφεται η μαθηματική θεμελίωση των αλγορίθμων αποικίας μυρμηγκιών. Στη συνέχεια προτείνεται ένα νέο μαθηματικό μοντέλο για την εξάτμιση της φερομόνης, με εμβάθυνση στη γεωμετρική και αλγεβρική σύγκλιση αντίστοιχα.

Στο *Κεφάλαιο 6* προτείνεται ένα γενικευμένο μαθηματικό μοντέλο διαχείρισης φερομόνης. Στη συνέχεια προτείνονται μαθηματικά μοντέλα φερομόνης, μέσω εξισώσεων διαφορών των κυρίων αλγορίθμων ACS και MMAS.

Στο *Κεφάλαιο 7* προτείνεται μια διαφοροποιημένη μορφή του αλγορίθμου Ant Colony System (DFACS) για την επίλυση των προβλημάτων δρομολόγησης οχημάτων (VRP). Στόχος μας είναι η εύρεση της αντικειμενικής συνάρτησης που περιγράφει το πρόβλημα και η μελέτη της συμπεριφοράς της, όταν μεταβάλλονται τα επίπεδα της αρχικής φερομόνης.

Στο *Κεφάλαιο 8* προτείνεται η εφαρμογή του αλγόριθμου DFACS για τη βέλτιστη λειτουργία ενός δικτύου πλοίων σε ένα πιλοτικό σύνολο

δεκατριών (13) λιμανιών (συμπεριλαμβανομένου του κεντρικού λιμανιού) της θάλασσας του Αιγαίου.

Στο *Κεφάλαιο 9* γίνεται εφαρμογή των αλγορίθμων αποικίας μυρμηγκιών (ACO) στα Κέντρα Ελέγχου Ηλεκτρικής Ενέργειας. Το πρόβλημα κατανομής ηλεκτρικού φορτίου είναι συνεχές. Οι κλασικοί αλγόριθμοι ACO εφαρμόζονται σε διακριτά προβλήματα. Προτείνεται η μέθοδος μετατροπής του συνεχούς σε διακριτό πρόβλημα, οπότε προσαρμόζεται κατάλληλα ο αλγόριθμος MMAS που ελαχιστοποιεί την αντικειμενική συνάρτηση που περιγράφει το πρόβλημα. Τέλος γίνεται σύγκριση των αποτελεσμάτων με γενετικούς αλγορίθμους.

Στο *Κεφάλαιο 10* λύνεται το πρόβλημα χωροθέτησης – κατανομής σε μια γραμμή με ένα πρωτότυπο αλγόριθμο αποικίας μυρμηγκιών τον up-ACS. Το πρόβλημα παρουσιάζει συνεχείς και διακριτές συνιστώσες. Η εφαρμογή του αλγορίθμου up-ACS παίρνει υπόψη του τα χαρακτηριστικά του προβλήματος με αποτέλεσμα να οδηγηθούμε σε καλές λύσεις.

Στο *Κεφάλαιο 11* προτείνεται η λύση του προβλήματος γραμμικής τοποθέτησης μηχανών, μιας εργοστασιακής μονάδας, με τη χρήση αλγορίθμου αποικίας μυρμηγκιών. Τέλος προτείνεται μια πρωτότυπη παρουσίαση του παραπάνω αλγορίθμου που οδηγεί στη βέλτιστη λύση του προβλήματος.

Παρατίθενται τρία Παραρτήματα που αφορούν τα ακόλουθα:

Το *Παράρτημα Ι* περιγράφει τον κώδικα σε περιβάλλον Delfi που οδήγησε την υλοποίηση του αλγορίθμου DFACS.

Στο *Παράρτημα ΙΙ* δίνεται ο ψευδοκώδικας του αλγορίθμου που χρησιμοποιήθηκε στη λύση της κατανομής του ηλεκτρικού φορτίου στα Κέντρα Ελέγχου Ενέργειας.

Στο *Παράρτημα ΙΙΙ* δίνεται ο ψευδοκώδικας του αλγορίθμου που χρησιμοποιήθηκε στη λύση του προβλήματος γραμμικής τοποθέτησης μηχανών.

ΚΕΦΑΛΑΙΟ 2

ΣΥΓΧΡΟΝΟΙ ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

2.1 Εισαγωγή

Η βελτιστοποίηση αποτελεί ένα βασικό θέμα συζητήσεων στην πληροφορική, στην τεχνητή νοημοσύνη, στην επιχειρησιακή έρευνα και σε άλλα σχετικά πεδία. Έξω από τις επιστημονικές κοινότητες, το νόημα της λέξης «βελτιστοποίηση» είναι μάλλον ανακριβές. Απλά σημαίνει «να γίνει κάτι καλύτερα». Ωστόσο, στα πλαίσια αυτής της διατριβής, η βελτιστοποίηση είναι η διαδικασία εύρεσης της βέλτιστης πιθανής λύσης σε ένα *πρόβλημα βελτιστοποίησης*, συνήθως, μέσα σε ένα δοσμένο χρονικό όριο. Με την σειρά του, ένα πρόβλημα βελτιστοποίησης είναι απλά ένα πρόβλημα για το οποίο υπάρχουν διάφορες πιθανές λύσεις, οι οποίες μάλιστα είναι αρκετά ποιοτικές. Με άλλα λόγια, ένα πρόβλημα βελτιστοποίησης είναι το πρόβλημα εκείνο για το οποίο έχει νόημα η σύγκριση και η αντίθεση των διαφορετικών υποψήφιων λύσεων του [Corne και άλλοι, 1999].

Τα τελευταία 40 χρόνια πολλοί ερευνητές έχουν μελετήσει το πρόβλημα [Reeves και άλλοι, 1993] της εύρεσης των βέλτιστων λύσεων σε προβλήματα τα οποία μπορούν να αναπαρασταθούν μαθηματικά ως μια συνάρτηση *μεταβλητών αποφάσεων*, υπό την παρουσία μερικών *περιορισμών*. Η μαθηματική έκφραση τέτοιων προβλημάτων είναι:

Ελαχιστοποίηση (minimize) της $f(x)$

η οποία υπόκειται στους περιορισμούς:

$$g_i(x) \geq b_i, \quad i = 1, \mathbf{K}, m \text{ και}$$

$$h_i(x) = c_j, \quad j = 1, \mathbf{K}, n.$$

Η μεταβλητή x είναι, ουσιαστικά, ένα διάνυσμα μεταβλητών αποφάσεων και οι συναρτήσεις $f(\cdot), g_i(\cdot), h_j(\cdot)$ είναι γενικής φύσεως. Στην πράξη, οι περιορισμοί ισότητας δεν είναι απόλυτα απαραίτητοι, καθώς μπορούν να αντικατασταθούν με ζευγάρια ανισώσεων. Αλλά συνήθως,

καλύτερα να είναι ρητοί. Εδώ, θεωρείται ένα πρόβλημα ελαχιστοποίησης, αλλά με τον ίδιο τρόπο, κάνοντας τις απαραίτητες αλλαγές, μπορεί να οριστεί ένα πρόβλημα μεγιστοποίησης.

Υπάρχουν πολλές ειδικές κατηγορίες τέτοιων προβλημάτων, ανάλογα με τους προτεινόμενους περιορισμούς στις, υπό μελέτη, συναρτήσεις και στις τιμές που μπορούν να πάρουν οι μεταβλητές αποφάσεων. Ίσως, η πιο γνωστή από αυτές τις κατηγορίες είναι αυτή, στην οποία οι περιορισμοί θέτουν τις $f(\cdot), g_i(\cdot), h_j(\cdot)$ να είναι γραμμικές συναρτήσεις μεταβλητών αποφάσεων, οι οποίες επιτρέπονται να είναι κλασματικές ή συνεχείς μεταβλητές. Με αυτό τον τρόπο, το πρόβλημα οδηγείται στον *γραμμικό προγραμματισμό*.

Μια βασική κατηγορία προβλημάτων βελτιστοποίησης είναι αυτή των *συνδυαστικών προβλημάτων*. Αυτός ο όρος χρησιμοποιείται σε προβλήματα των οποίων οι μεταβλητές αποφάσεων είναι διακριτές. Για παράδειγμα, όταν η λύση είναι ένα σύνολο ή μια ακολουθία από ακέραιους ή άλλα διακριτά αντικείμενα. Η διαδικασία εύρεσης βέλτιστων λύσεων σε τέτοιου είδους προβλήματα είναι γνωστή ως *συνδυαστική βελτιστοποίηση*.

Μερικά παραδείγματα προβλημάτων συνδυαστικής βελτιστοποίησης είναι:

Παράδειγμα 2.1: **Το πρόβλημα εκχώρησης**. Ένα σύνολο από n ανθρώπους είναι διαθέσιμο για να φέρει εις πέρας n καθήκοντα. Αν ένα άτομο i εκτελεί j καθήκοντα, τότε το κόστος είναι c_{ij} μονάδες. Να βρεθεί μια εκχώρηση $\{\pi_1, \mathbf{K}, \pi_n\}$ η οποία να ελαχιστοποιεί το άθροισμα:

$$\sum_{i=1}^{n} c_{i\pi_i}$$

Η λύση προκύπτει από την μετάθεση $\{\pi_1, \mathbf{K}, \pi_n\}$ των αριθμών $\{1, \mathbf{K}, n\}$.

Παράδειγμα 2.2: **Το 0-1 πρόβλημα του σάκου**. Ένα σύνολο από n αντικείμενα είναι διαθέσιμο να πακεταριστεί σε ένα σάκο με χωρητικότητα C μονάδων. Το αντικείμενο i έχει αξία υ_i και χρησιμοποιεί c_i μονάδες χωρητικότητας. Να προσδιοριστεί το υποσύνολο I των αντικειμένων που θα πακεταριστεί για να μεγιστοποιηθεί το άθροισμα:

$$\sum_{i \in I} \upsilon_i$$

τέτοιο ώστε $\sum_{i\in I} c_i \le C$.

Η λύση είναι το υποσύνολο $I \subseteq \{1, \mathbf{K}, n\}$.

Παράδειγμα 2.3: **Το πρόβλημα επικάλυψης συνόλου**. Μια οικογένεια από m υποσύνολα περιέχει n αντικείμενα έτσι ώστε το υποσύνολο S_i να περιέχει $n_i\ (\le n)$ αντικείμενα. Να επιλεχθούν $k\ (< m)$ υποσύνολα $\{S_{i_1}, \mathbf{K}, S_{i_k}\}$ τέτοια ώστε $\left|\mathbf{U}_{j=1}^{k} S_{i_j}\right| = n$ έτσι ώστε να ελαχιστοποιηθεί το άθροισμα:

$$\sum_{j=1}^{k} c_{i_j}$$

όπου c_i είναι το κόστος επιλογής του υποσυνόλου S_i.
Η λύση είναι η οικογένεια των υποσυνόλων $\{S_{i_1}, \mathbf{K}, S_{i_k}\}$.

Παράδειγμα 2.4: **Το πρόβλημα δρομολόγησης οχημάτων**. Μια αποθήκη έχει m οχήματα διαθέσιμα για να παραδοθούν σε n πελάτες. Η χωρητικότητα του οχήματος k είναι C_k μονάδες, ενώ ο πελάτης i χρειάζεται c_i μονάδες. Η απόσταση μεταξύ του πελάτη i και του πελάτη j είναι d_{ij}. Κανένα όχημα δεν μπορεί να ταξιδέψει σε απόσταση μεγαλύτερη από D μονάδες. Να κατανεμηθούν οι πελάτες στα οχήματα και να βρεθεί η σειρά με την οποία κάθε όχημα επισκέπτεται τους πελάτες του έτσι ώστε να ελαχιστοποιηθεί το άθροισμα:

$$\sum_{k=1}^{m} \sum_{i=0}^{n_k} d_{\pi_{i,k}, \pi_{i+1,k}}$$

τέτοιο ώστε

$$\sum_{i=1}^{n_k} c_{\pi_{i,k}} \le C_k, \qquad k = 1, \mathbf{K}, m$$

$$\sum_{i=0}^{n_k} d_{\pi_{i,k}, \pi_{i,k}} \le D, \qquad k = 1, \mathbf{K}, m$$

$$\sum_{k=1}^{m} n_k = n$$

Εδώ, το όχημα k επισκέπτεται n_k πελάτες, και η λύση περιγράφεται από την μετάθεση $\{\pi_{1,1},\mathbf{K},\pi_{n_1,1},\mathbf{K}\pi_{1,m},\mathbf{K},\pi_{n_m,m}\}$ των αριθμών $\{1,\mathbf{K},n\}$ η οποία διαμερίζεται από τους αριθμούς $\{n_k\}$. Επίσης η αποθήκη αντιπροσωπεύεται από τους πελάτες $\pi_{0,k}$ και $\pi_{n_k+1,k}$ για κάθε k.

Πρέπει να διευκρινιστεί ότι σε αυτά τα παραδείγματα χρησιμοποιείται ο όρος «πρόβλημα» με τη γενική του έννοια. Σε μια πραγματική περίπτωση, θα υπάρχει ένα συγκεκριμένο πρόβλημα στο οποίο οι μεταβλητές που θα το περιγράφουν έχουν πραγματικές αριθμητικές τιμές. Συνηθίζεται να χρησιμοποιείται ο όρος *στιγμιότυπο* (instance) για την διάκριση μεταξύ της συγκεκριμένης και της γενικής περίπτωσης.

Τα προβλήματα βελτιστοποίησης καλύπτουν τα πεδία της επιστήμης και βιομηχανίας αντίστοιχα. Υπάρχουν πολλοί τρόποι για να χαραχθεί ένα δίκτυο γραμμών μεταφοράς ή ένα δίκτυο καλωδίων ή ένα δίκτυο ροής πληροφοριών. Όμως, τίθεται το ερώτημα: «Ποιος από αυτούς είναι ο πιο αξιόπιστος;», [Corne και άλλοι, 1999]. Η απάντηση μπορεί να είναι εύκολη επειδή υπάρχουν λίγες πιθανές υποψήφιες λύσεις που μπορεί κάποιος να γνωρίζει. Στην περίπτωση ύπαρξης πολλών υποψήφιων λύσεων, που οδηγούν σε εξαντλητική έρευνα, είναι απαραίτητη η κατανόηση του προβλήματος που θα οδηγήσει στην εύρεση της βέλτιστης λύσης ή σε μια κατεύθυνση προς αυτή.

Πολλά προβλήματα βελτιστοποίησης είναι *«δύσκολα»* προβλήματα (hard problems). Τέτοια είναι τα προβλήματα που συναντώνται στον πραγματικό κόσμο, στη βιομηχανία και στην επιστήμη. Στην επιστήμη της πληροφορικής ένα «δύσκολο» πρόβλημα έχει την δική του σημασία. Συγκεκριμένα, ένα «δύσκολο» πρόβλημα είναι ένα πρόβλημα για το οποίο δεν υπάρχει καμία εγγύηση για την εύρεση της βέλτιστης λύσης του σε ένα αποδεκτό χρονικό διάστημα [Harel, 1987]. Στην πράξη, τα αποτελέσματα της έρευνας για την επίλυση «δύσκολων» προβλημάτων δεν είναι τόσο αποθαρρυντικά σε σχέση με τον ορισμό τους. Αυτό οφείλεται κυρίως στη χρήση *προσεγγιστικών* μεθόδων (approximate methods). Μια προσεγγιστική μέθοδος είναι ένας αλγόριθμος που χρησιμοποιείται για την εύρεση λύσεων σε προβλήματα βελτιστοποίησης. Ο αλγόριθμος, αν και είναι γρήγορος, δεν εγγυάται ότι η λύση που θα βρει θα είναι και η βέλτιστη δυνατή. Μερικές προσεγγιστικές μέθοδοι, όταν χρησιμοποιούνται για συγκεκριμένα προβλήματα βελτιστοποίησης, μπορούν να βρουν την βέλτιστη λύση πολύ γρήγορα.

Ένα συνηθισμένο και πρακτικό παράδειγμα βελτιστοποίησης είναι το πρόβλημα του περιοδεύοντος πωλητή (TSP) [Lawler και άλλοι, 1985]. Το εποπτικό φυσικό πρόβλημα του προβλήματος (TSP) είναι: ένας περιοδεύον πωλητής ξεκινά από μια πόλη (βάση) και πρέπει να επισκεφτεί έναν ορισμένο αριθμό πόλεων, μια φορά την κάθε μια, και στη συνέχεια να επιστρέψει στην βάση του. Ζητείται να βρεθεί η σειρά με την οποία πρέπει να τις επισκεφτεί, ώστε το κόστος που θα προκύψει για τον περιοδεύοντα πωλητή να είναι ελάχιστο. Το πρόβλημα (TSP) διατυπώνεται, διαφορετικά, και ως πρόβλημα εύρεσης της συντομότερης διαδρομής. Γενικά πρόκειται για ένα «δύσκολο» πρόβλημα που οι κλασικές διαδικασίες βελτιστοποίησης δεν εγγυώνται τα καλύτερα αποτελέσματα. Ωστόσο, η πρόοδος των μεθόδων βελτιστοποίησης αύξησαν, αποτελεσματικά, τη δυνατότητα εύρεσης βέλτιστης λύσης σε «δύσκολα» προβλήματα.

2.2 Στοχαστικές επαναληπτικές μέθοδοι

Οι μέθοδοι βελτιστοποίησης είναι τα κύρια εργαλεία που χρησιμοποιούνται προκειμένου να επιλυθούν τα προβλήματα βελτιστοποίησης. Υπάρχουν δυο βασικές και πολύ χρήσιμες μέθοδοι βελτιστοποίησης:

- Ειδικές μέθοδοι, που σχεδιάζονται κατάλληλα για ορισμένες κατηγορίες προβλημάτων.

- Γενικές μέθοδοι, που εφαρμόζονται σε όλα, σχεδόν, τα προβλήματα βελτιστοποίησης..

Στην παρούσα διατριβή το ενδιαφέρον επικεντρώνεται στις *στοχαστικές επαναληπτικές μεθόδους αναζήτησης*, οι οποίες ανήκουν στην κατηγορία των γενικών μεθόδων βελτιστοποίησης. Οι μέθοδοι αυτοί ακολουθούν συνήθως το μοντέλο του αλγορίθμου που δείχνεται στο Σχήμα 2-1. Οι στοχαστικές επαναληπτικές μέθοδοι αναζήτησης υπερτερούν των άλλων μεθόδων για τρεις λόγους:

- Είναι πολύ απλές στην υλοποίηση. Απαιτούνται μόνο μερικές γραμμές ψευδοκώδικα ή κανονικού κώδικα για να περιγραφθούν τα ουσιώδη στοιχεία των περισσότερων τέτοιων μεθόδων. Για παράδειγμα, ο Price δίνει έμφαση στο πόσο συμπαγής είναι ο Διαφορικός Εξελικτικός Αλγόριθμος (Differential Evolution Algorithm) ο οποίος υπερτερεί κατά πολύ έναντι μερικών πιο

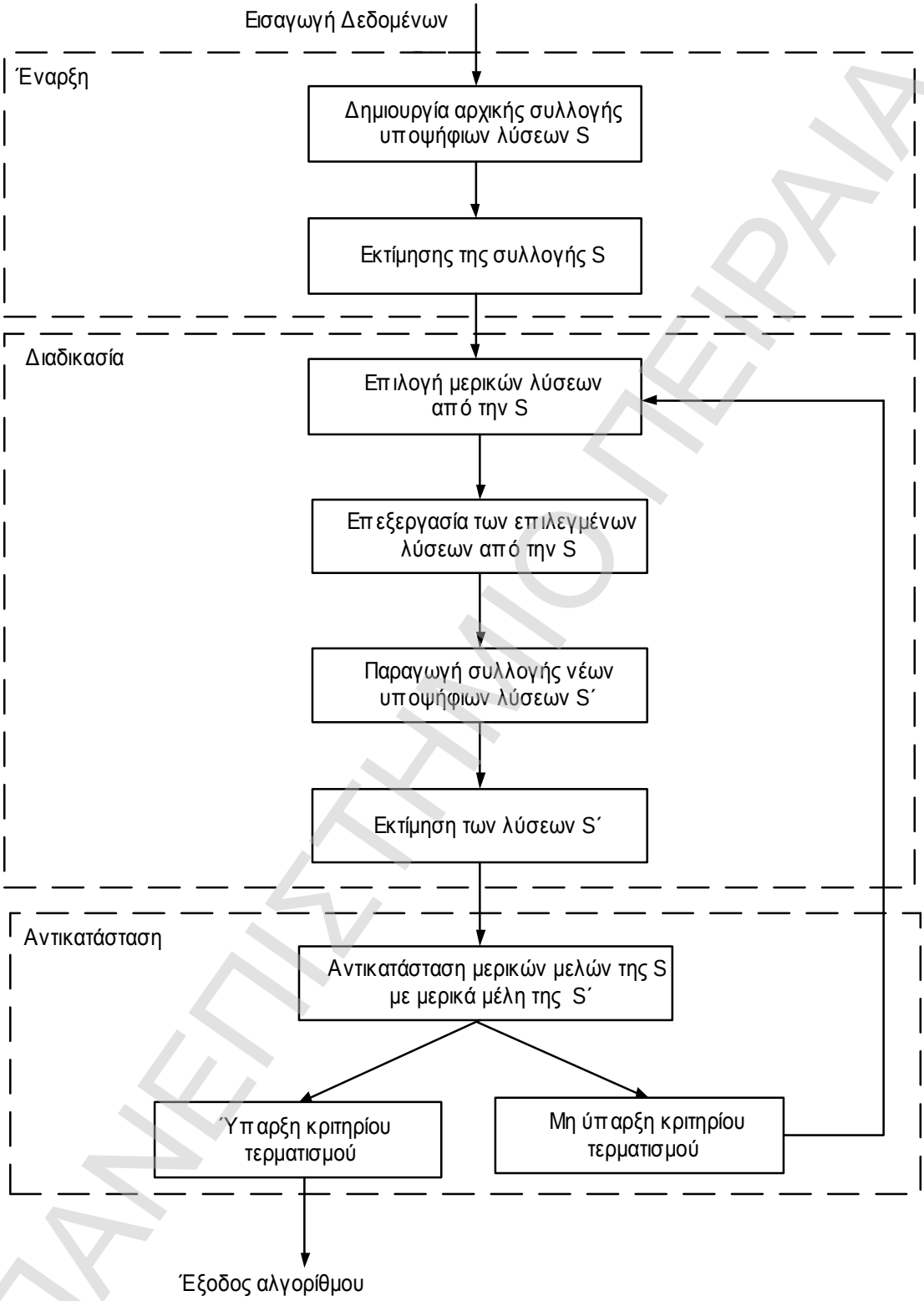
πρωτοποριακών μεθόδων, τουλάχιστον, για μερικά είδη προβλημάτων.

- Είναι πολύ γενικές. Δεν υπάρχει κάποια πρωταρχική απαίτηση για τα είδη των προβλημάτων στα οποία απευθύνονται. Για παράδειγμα, δεν είναι αναγκαίο να είναι διαφορίσιμη η συνάρτηση παραμέτρων με πραγματικές τιμές ή οι παράμετροι να περιγράφονται από κάποια ειδική γλώσσα περιορισμών.

- Έχουν αποδειχτεί πολύ πετυχημένες στην εύρεση καλών και γρήγορων λύσεων σε δύσκολα προβλήματα βελτιστοποίησης σε πολλές πρακτικές και επιστημονικές εφαρμογές. Αυτή η επιτυχία οφείλεται σε δύο, κυρίως, λόγους.

  i. Όταν χρησιμοποιούνται, οι στοχαστικές επαναληπτικές μέθοδοι αναζήτησης μπορούν να εφαρμοστούν σε οποιοδήποτε πρόβλημα βελτιστοποίησης με μικρή ή ακόμα μηδαμινή προσπάθεια από αυτή που απαιτείται με την υλοποίηση του κώδικα, για την εύρεση υποψήφιων λύσεων του προβλήματος. Επίσης, η επιτυχία οφείλεται στο γεγονός ότι με τόσο μικρή ανάπτυξη και υλοποίηση τα αποτελέσματα είναι συχνά εντυπωσιακά.

  ii. Αυτές οι τεχνικές παρέχουν τα πλαίσια εργασίας που μπορούν εύκολα και απόλυτα να αντιμετωπιστούν από ειδικές ευρεστικές μεθόδους, προκειμένου να εκμεταλλευτούν την γνώση, σχετικά με το πρόβλημα.



Σχήμα 2-1: Γενικευμένες στοχαστικές επαναληπτικές μέθοδοι αναζήτησης. Διάγραμμα Δραστηριοτήτων της μεθόδου.

Οι στοχαστικές επαναληπτικές μέθοδοι αναζήτησης στηρίζονται στην βασική ιδέα του *«δημιουργώ και ελέγχω»* (“generate and test”). Κατά κάποιο τρόπο, δημιουργείται μια υποψήφια λύση για το δοσμένο πρόβλημα. Αν βρεθεί ότι δεν είναι αρκετά καλή, τότε η διαδικασία επαναλαμβάνεται, και κάθε φορά δημιουργείται μια διαφορετική υποψήφια λύση για έλεγχο. Το κλειδί της επαναληπτικής τεχνικής «δημιουργώ και ελέγχω» είναι η *τυχαία αναζήτηση* (random search). Σε κάθε επανάληψη, μια νέα τυχαία λύση προκύπτει. Εν τούτοις, μια τεράστια ποικιλία από μεθόδους βελτιστοποίησης έχουν ερευνηθεί οι οποίες ακολουθούν την βασική ιδέα «δημιουργώ και ελέγχω». Όμως, χρησιμοποιούν περισσότερο ενδιαφέρουσες και αποτελεσματικές μεθόδους για την δημιουργία νέων υποψήφιων λύσεων. Σε αυτές τις μεθόδους, το κλειδί είναι η χρήση της πληροφορίας από τις προηγούμενες λύσεις που θα οδηγήσει στη γέννηση νέων υποψήφιων λύσεων. Εφόσον, σε ένα πραγματικό πρόβλημα, ο αριθμός των δυνατών υποψήφιων λύσεων είναι κατά πολύ μεγαλύτερος από τον αναμενόμενο αριθμό που θα δημιουργηθεί και θα ελεγχθεί σε ένα λογικό χρόνο, η διαδικασία δημιουργίας υποψήφιων λύσεων του προβλήματος θα είναι, σχεδόν, επιτυχής.

Για να σχεδιαστεί μια μέθοδος βελτιστοποίησης, αρκεί να σχεδιαστεί ένας τρόπος που να απαντά στην παρακάτω ερώτηση: πώς μπορεί να χρησιμοποιηθεί η πληροφορία που συγκεντρώθηκε από τις προηγούμενες υποψήφιες λύσεις προκειμένου να δημιουργηθούν νέες; Στην τυχαία αναζήτηση δεν χρησιμοποιείται η πληροφορία από προηγούμενες λύσεις και συνεπώς η αποτελεσματικότητα της είναι πολύ μικρή. Στις σημερινές, όμως, σύγχρονους μεθόδους βελτιστοποίησης, χρησιμοποιούνται τρεις, αποδεδειγμένα, επιτυχημένες κατευθύνσεις που ανταποκρίνονται στην παραπάνω ερώτηση, οι οποίες είναι:

- Κατεύθυνση 1: Οι νέες υποψήφιες λύσεις διαφέρουν ελάχιστα από τις προηγούμενες υποψήφιες λύσεις.

- Κατεύθυνση 2: Οι νέες υποψήφιες λύσεις δημιουργούνται, συνδυάζοντας παραλλαγές δυο ή παραπάνω υπαρχόντων υποψήφιων λύσεων.

- Κατεύθυνση 3: Οι «γονείς» από τους οποίους οι νέες υποψήφιες λύσεις παράγονται (από τις κατευθύνσεις 1 ή 2), επιλέγονται από προηγούμενες υποψήφιες λύσεις μέσω μιας στοχαστικής και ανταγωνιστικής τεχνικής, η οποία ευνοεί τις λύσεις με την καλύτερη απόδοση.

Οι ελάχιστα διαφοροποιημένες λύσεις που αναφέρονται στην Κατεύθυνση 1, υλοποιούνται από την, λεγόμενη, *γειτονιά* (neighborhood) ή τους *τελεστές μεταβολής* (mutation operators) ή τους *τελεστές κίνησης* (move operators). Για παράδειγμα, απευθυνόμενοι σε ένα στιγμιότυπο του προβλήματος του περιοδεύοντος πωλητή (TSP), τότε μπορεί να κωδικοποιηθεί μια υποψήφια λύση, σαν μια μετάθεση (permutation) των προς επίσκεψη πόλεων. Η *γειτονιά* σε αυτή την περίπτωση μπορεί να είναι η μετάθεση των θέσεων, τυχαία, επιλεγμένων πόλεων.

Οι τελεστές που παράγουν νέες υποψήφιες λύσεις από δυο ή περισσότερες υπάρχουσες λύσεις ονομάζονται *τελεστές διασταύρωσης* (crossover operators) ή *συνδυαστικοί τελεστές* (recombination operators) ή *τελεστές συγχώνευσης* (merge operators). Μερικά παραδείγματα τέτοιων τελεστών εκτελούν ενέργειες του τύπου «κόβω και ενώνω» (“cut and splice”). Άλλα παραδείγματα περιλαμβάνουν περισσότερο πρωτοποριακές τεχνικές, στις οποίες δυο ή παραπάνω «γονικές» υποψήφιες λύσεις εισάγονται σε μια αυτο-συντηρούμενη αλγοριθμική διαδικασία αναζήτησης, η οποία με την σειρά της παράγει μια ή περισσότερες υποψήφιες λύσεις – απογόνους.

Τελικά, μια τυχαία αλλά ανταγωνιστική τεχνική χρησιμοποιείται για να αποφασιστεί ποιος θα είναι ο «γονέας». Σύμφωνα με την τεχνική αυτή, όσο πιο ανταγωνιστική είναι μια υποψήφια λύση, τόσο πιο πιθανό είναι να επιλεγεί για να γίνει «γονέας» και έτσι να αποτελέσει τη βάση (ή τουλάχιστον μέρος) της νέας υποψήφιας λύσης.

Γνωστές μέθοδοι βελτιστοποίησης, που χρησιμοποιούν τουλάχιστον μια από τις παραπάνω βασικές κατευθύνσεις, είναι:

- Simulated Annealing [Aarts & Korst, 1989],
- Tabu Search [Glover, 1989, 1990],
- Genetic Algorithms [Holland, 1975].

Οι παραπάνω μέθοδοι περιλαμβάνουν μια μεγάλη ποικιλία από βοηθητικά χαρακτηριστικά, για τον έλεγχο της πορείας της αναζήτησης και την προσαρμογή της συμπεριφοράς των μεθόδων κατά την διάρκεια της αναζήτησης.

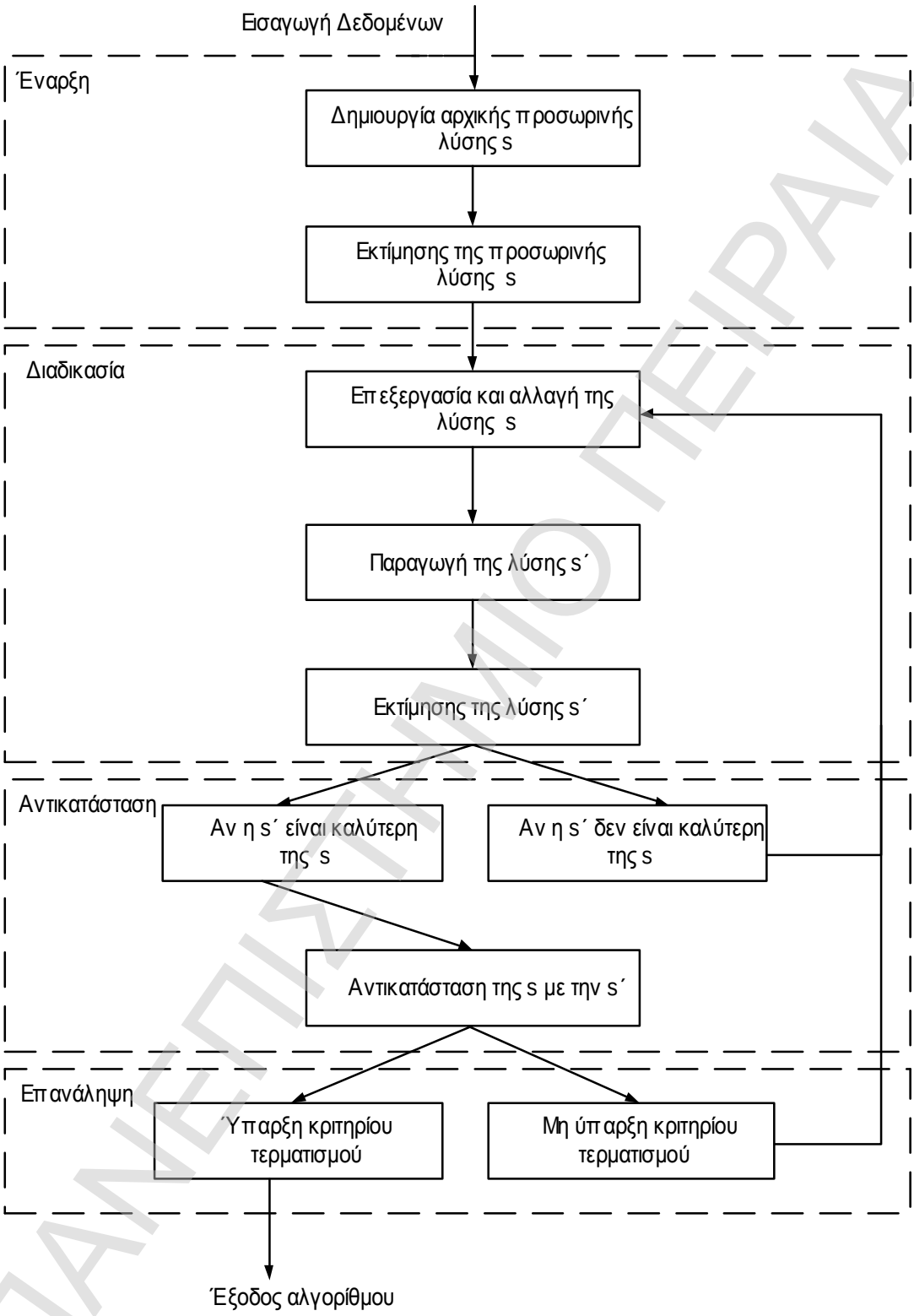
2.3 Νέες κατευθύνσεις βελτιστοποίησης

Οι υπάρχουσες επιτυχημένες μέθοδοι στοχαστικής επαναληπτικής βελτιστοποίησης ανήκουν σε δυο γενικές κατηγορίες:

- *Τοπική αναζήτηση* και
- *Αναζήτηση βασισμένη στον πληθυσμό*.

Στην *τοπική αναζήτηση* υπάρχει μια προσωρινή τρέχουσα λύση μέχρι να ερευνηθούν οι γειτονικές της λύσεις (νέες υποψήφιες λύσεις που διαφέρουν ελάχιστα από αυτήν) προκειμένου να βρεθούν καλύτερες ποιοτικά λύσεις. Περιστασιακά, μια από αυτές τις γειτονικές λύσεις γίνεται η νέα προσωρινή τρέχουσα λύση και στη συνέχεια οι δικές τις γειτονικές λύσεις ερευνούνται κ.ο.κ. Το απλούστερο παράδειγμα της μεθόδου τοπικής αναζήτησης είναι ο αλγόριθμος Hillclimbing (αλλιώς ονομάζεται τυχαίας μεταβολής Hillclimbing (random mutation Hillclimbing) ή στοχαστικός hillclimbing (stochastic Hillclimbing)). Στο Σχήμα 2-2 περιγράφονται τα βήματα του αλγορίθμου.

Ο αλγόριθμος Hillclimbing, ουσιαστικά, επαναλαμβάνει την Κατεύθυνση 1, παράγοντας συνεχώς νέες υποψήφιες λύσεις οι οποίες διαφέρουν ελάχιστα από τις τρέχουσες προσωρινές λύσεις. Επίσης, περιλαμβάνει μια απλή αλλά αποτελεσματική παραλλαγή της Κατεύθυνσης 3: η λύση που αποτελεί την βάση για την δημιουργία των νέων υποψήφιων λύσεων είναι πάντα η βέλτιστη που έχει βρεθεί μέχρι στιγμής (στην πραγματικότητα, είναι η λύση που η απόδοση της είναι η καλύτερη μέχρι στιγμής). Περισσότερο πρωτοποριακές μέθοδοι τοπικής αναζήτησης βελτίωσαν τον αλγόριθμο Hillclimbing, κάνοντας πιο προσεκτική και πιο έξυπνη την διαδικασία δημιουργίας των νέων υποψήφιων λύσεων. Για παράδειγμα, στη μέθοδο Simulated Annealing, η διαφορά βρίσκεται στο βήμα 3: μερικές φορές, αποδεχόμαστε την λύση s΄ ως την νέα προσωρινή λύση, αν και μπορεί να είναι χειρότερη από την s. Χωρίς αυτή την διευκόλυνση, ο αλγόριθμος Hillclimbing «κολλάει» σε τοπικά βέλτιστα, δηλαδή σε λύσεις των οποίων οι γειτονικές τους λύσεις είναι χειρότερες από την τρέχουσα προσωρινή λύση ή σε περιοχές του χώρου λύσεων όπου αν και υπάρχουν αξιοσημείωτες λύσεις με ισότιμη απόδοση και ποιότητα, ωστόσο πολύ λίγες ή καμία από αυτές τις λύσεις δεν έχει μια αποδοτική και ανταγωνιστική γειτονιά υποψήφιων λύσεων. Με αυτή την διευκόλυνση, η μέθοδος της Simulated Annealing, καθώς και άλλες μέθοδοι τοπικής αναζήτησης, μπορούν μερικές φορές (αλλά σίγουρα όχι πάντα) να αποφύγουν τέτοιες δυσκολίες.



Σχήμα 2-2: Ο απλός αλγόριθμος Hillclimbing. Διάγραμμα δραστηριοτήτων του αλγορίθμου.

Στην *αναζήτηση βασισμένη στον πληθυσμό*, η έννοια της μοναδικής προσωρινής λύσης αντικαθίσταται από την ύπαρξη ενός πληθυσμού από διαφορετικές προσωρινές λύσεις. Οι νέες λύσεις δημιουργούνται, αφού πρώτα έχουν επιλεγεί στοιχεία από αυτόν τον πληθυσμό για να είναι «γονείς», κάνοντας αλλαγές σε αυτούς τους γονείς για να παραχθούν οι απόγονοι. Η αναζήτηση βασισμένη στον πληθυσμό στηρίζεται στην Κατεύθυνση 2. Το γεγονός ότι υπάρχει μια συλλογή από προσωρινές τρέχουσες λύσεις, κι όχι μόνο μια, δίνει την δυνατότητα να χρησιμοποιούνται δυο ή περισσότερες λύσεις από αυτή τη συλλογή για να αποτελέσουν την βάση για τις νέες υποψήφιες λύσεις που θα δημιουργηθούν. Επίσης, η εισαγωγή του πληθυσμού λύσεων δημιουργεί νέες δυνατότητες. Για παράδειγμα, η Κατεύθυνση 3 είναι τώρα ανοιχτή σε πιο ενδιαφέρουσες αντιλήψεις για το πια τεχνική θα επιλεγεί προκειμένου να αποφασιστεί ποιες λύσεις θα είναι οι «γονείς». Επίσης, απαιτείται μια τεχνική για την διατήρηση του πληθυσμού, όταν μια ή περισσότερες υποψήφιες λύσεις έχουν παραχθεί. Δηλαδή, προκειμένου να διατηρηθεί ο πληθυσμός σε ένα σταθερό μέγεθος (που είναι πάντα το ζητούμενο), πρέπει μερικές λύσεις από τον πληθυσμό να απορριφθούν για να κάνουν χώρο για μερικές ή όλες τις νέες υποψήφιες λύσεις που παρήχθηκαν από τις διαδικασίες μεταβολής ή συνδυασμού. Παρατηρείται, ότι μια τεχνική επιλογής συνδυασμένη με μια τεχνική διατήρησης πληθυσμού καταλήγει σε μια γενική τεχνική για την υλοποίηση μιας διαφοροποίησης της Κατεύθυνσης 3. Ενώ η τεχνική επιλογής περιγράφει τις λεπτομέρειες πώς να επιλεγούν οι υποψήφιες λύσεις από τον ίδιο τον πληθυσμό, η τεχνική διατήρησης πληθυσμού επηρεάζει οτιδήποτε υπάρχει μέσα στον πληθυσμό, και επομένως μπορεί και να επιλεγεί.

Υπάρχουν, τόσοι πολλοί αλγόριθμοι βασισμένοι στον πληθυσμό βελτιστοποίησης και διαφορετικές εκδοχές τους, όσοι είναι οι τρόποι που μπορούν να αντιμετωπιστούν τα παραπάνω. Το βασικό χαρακτηριστικό αυτών των αλγορίθμων, που είναι πιο γνωστοί και ως εξελικτικοί αλγόριθμοι (evolutionary algorithms), είναι οι πολλοί τρόποι που έχουν χρησιμοποιηθεί για να υλοποιηθούν οι Κατευθύνσεις 1, 2 και 3. Μια πληθώρα ερωτημάτων που περιβάλλουν αυτά τα ζητήματα μόλις αρχίζουν να απαντώνται από την επιστημονική κοινότητα. Για παράδειγμα, είναι καλύτερο να χρησιμοποιείται τοπική αναζήτηση ή αναζήτηση βασισμένη στον πληθυσμό για ένα δοσμένο πρόβλημα; Τι είδους τεχνική επιλογής θα λειτουργήσει καλύτερα; Αν έχει δοθεί το μέγεθος του πληθυσμού, μπορεί να προβλεφθεί σε πόσο καιρό θα βρει η μέθοδος αναζήτησης βασισμένη στον πληθυσμό, την καλύτερη λύση; Τα αποτελέσματα, όμως, μιας έντονα αναπτυσσόμενης έρευνας πάνω σε τέτοια ερωτήματα, αποκάλυψαν ότι είναι, πιθανόν αφελέστατα, να περιμένουμε οριστικές και σαφείς

απαντήσεις. Ωστόσο, υπάρχει μια αξιοσημείωτη εξάρτηση στο πρόβλημα, δηλαδή η μέθοδος που εφαρμόζεται, άψογα, σε ένα συγκεκριμένο πρόβλημα βελτιστοποίησης μπορεί να μην εφαρμόζεται το ίδιο καλά σε ένα άλλο, σχετικά παρόμοιο, πρόβλημα. Μια συνέπεια αυτής της γενικής παρατήρησης, που έχει ειδικό ενδιαφέρον, είναι ότι δικαιολογεί, μάλιστα ενθαρύνει, την έρευνα για νέες μεθόδους. Πράγματι, τέτοιες νέες μέθοδοι έχουν αναδυθεί τα πρόσφατα χρόνια, καθώς οι ερευνητές, παγκοσμίως, έχουν ερευνήσει τις πολλές ερωτήσεις που περιβάλλουν την τοπική και την βασισμένη στον πληθυσμό αναζήτηση.

2.4 Μετά- ευρεστικοί αλγόριθμοι βελτιστοποίησης αποικίας μυρμηγκιών

Τα προβλήματα συνδυαστικής βελτιστοποίησης έχουν μεγάλο ενδιαφέρον, επειδή είναι εύκολα να προσδιοριστούν αλλά δύσκολα να επιλυθούν. Πολλά από τα προβλήματα που εμφανίζονται στις εφαρμογές είναι NP-hard προβλήματα, δηλαδή προβλήματα για τα οποία πιστεύεται ότι δεν μπορούν να βρεθούν βέλτιστες λύσεις στα πλαίσια ενός πολυωνυμικού υπολογιστικού χρονικού ορίου. Για αυτό το λόγο, και προκειμένου να λυθεί ένα πρόβλημα, πρέπει να χρησιμοποιηθούν προσεγγιστικές μέθοδοι οι οποίες θα δώσουν σχεδόν βέλτιστες λύσεις σε ένα σχετικά σύντομο χρονικό διάστημα. Οι αλγόριθμοι αυτού του τύπου ονομάζονται, *ευρεστικοί* (heuristics). Συνήθως χρησιμοποιούν ειδική πληροφορία σχετικά με το πρόβλημα για να δημιουργήσουν ή να βελτιώσουν τις λύσεις.

Πρόσφατα, οι ερευνητές επικέντρωσαν το ενδιαφέρον τους σε μια νέα κατηγορία αλγορίθμων, που λέγονται *μετά- ευρεστικοί* (meta-heuristics). Ένας *μετά- ευρεστικός* αλγόριθμος είναι ένα σύνολο από αλγοριθμικές έννοιες που μπορούν να χρησιμοποιηθούν για να προσδιορίσουν ευρεστικές μεθόδους κατάλληλες για ένα ευρύ σύνολο από διαφορετικά προβλήματα. Η χρήση των μετα-ευρεστικών αλγορίθμων έχει αυξήσει σημαντικά την ικανότητα εύρεσης πολύ καλών, ποιοτικά, λύσεων, σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης, σε ένα λογικό χρονικό πλαίσιο.

Μια κατηγορία, επιτυχημένων, μετα-ευρεστικών αλγορίθμων εμπνέονται από τη συμπεριφορά των πραγματικών μυρμηγκιών [Dorigo και άλλοι, 1996]. Ξεκινώντας με τον αλγόριθμο Ant System [Dorigo, 1992], ένας αριθμός από αλγοριθμικές προσεγγίσεις, βασισμένες στις ίδιες βασικές αρχές, αναπτύχθηκαν και εφαρμόστηκαν, με αξιοσημείωτη επιτυχία, σε μια ποικιλία προβλημάτων συνδυαστικής βελτιστοποίησης, τόσο σε θεωρητικές

όσο και σε πραγματικές εφαρμογές. Σε αυτή τη διατριβή, θα μελετηθεί η βελτιστοποίηση της αποικίας μυρμηγκιών (ACO), η οποία είναι ένα μετα-ευρεστικό πλαίσιο εργασίας που καλύπτει την παραπάνω αλγοριθμική προσέγγιση. Οι ACO μετα- ευρεστικοί αλγόριθμοι έχουν προταθεί σαν ένα κοινό εργαλείο για τις υπάρχουσες εφαρμογές και αλγοριθμικές παραλλαγές μιας ποικιλίας από αλγορίθμους μυρμηγκιών. Οι αλγόριθμοι που ταιριάζουν με αυτό το πλαίσιο εργασίας θα ονομάζονται, από δω και πέρα, αλγόριθμοι ACO (ACO algorithms).

ΚΕΦΑΛΑΙΟ 3

ΚΟΙΝΩΝΙΚΑ ΕΝΤΟΜΑ ΚΑΙ ΟΙ ΣΥΛΛΟΓΙΚΕΣ ΤΟΥΣ ΣΥΜΠΕΡΙΦΟΡΕΣ

3.1 Εισαγωγή

Τα έντομα που ζουν με την κοινωνική μορφή της αποικίας: τα μυρμήγκια, οι μέλισσες, οι σφήκες και οι τερμίτες συναρπάζουν, για πολλά χρόνια, τους φυσιοδίφες, τους επιστήμονες και τους ανθρώπους, με την κοινωνική δομή τους και τον τρόπο οργάνωσης της ζωής τους [Bonabeau, Dorigo, Thelaulaz, 1999]. «Τι είναι αυτό που κυβερνά εδώ; Τι είναι αυτό που θέτει εντολές, προβλέπει το μέλλον, διαμορφώνει σχέδια και συντηρεί την ισορροπία;», έγραψε ο Maeterlinck (1927). Κάθε έντομο έχει καθορισμένη δομή. Η συμμετοχή του στην αποικία συμβάλει στην οργανωμένη μορφή της, ενώ η συνεχής ολοκλήρωση όλων των δραστηριοτήτων των εντόμων δεν απαιτεί την ύπαρξη «αρχηγού».

Για παράδειγμα, τα μυρμήγκια Leafcutter (Atta) κόβουν φύλλα από τα φυτά και τα δένδρα για να αναπτύξουν μύκητες. Οι εργάτες – μυρμήγκια προμηθεύονται τα φύλλα εκατοντάδες μέτρα μακριά από τη φωλιά τους, οργανώνοντας, κυριολεκτικά, εθνικές οδούς προς και από τις προμηθευτικές τους περιοχές [Holldobler, B. και E. O. Wiegon, 1990]. Τα μυρμήγκια έχουν αποικίσει κάθε μέρος του πλανήτη μας και αποτελούν μία από τις πιο οργανωμένες μορφές ζωής.

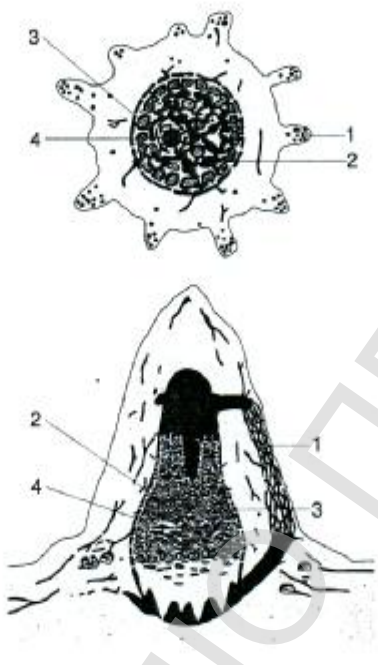
Σε μία αποικία κοινωνικών εντόμων, ένας εργαζόμενος, συνήθως, δεν εκτελεί όλους τους στόχους, αλλά, μάλλον, ειδικεύεται σε ένα σύνολο στόχων, σύμφωνα με την μορφολογία, την ηλικία ή τις ευκαιρίες του. Αυτό το τμήμα της εργασίας, με το οποίο οι διαφορετικές δραστηριότητες εκτελούνται ταυτόχρονα από τις ομάδες ειδικευμένων ατόμων, θεωρείται αποδοτικότερο από το να εκτελούνται διαδοχικά από ανειδίκευτα άτομα [Jeanne, R. L., 1986].

Οι μέλισσες (Appis Mellifica) χτίζουν σειρά παράλληλων κόμβων δημιουργώντας αλυσίδες που προκαλούν μια τοπική αύξηση στη θερμοκρασία [Darchen, R., 1959]. Οι κόμβοι κεριών μπορούν να διαμορφωθούν ευκολότερα χάρη σε αυτή την αύξηση θερμοκρασίας. Με τις

συνδυασμένες δυνάμεις των ατόμων στις αλυσίδες, οι κόμβοι κεριών μπορούν να είναι και να γίνονται παράλληλα ο ένας προς τον άλλο. Κάθε κόμβος της αλυσίδας οργανώνεται σε ένα πλήθος ομόκεντρων δακτυλιδιών, γύρης και μελιού. Οι πηγές τροφίμων αξιοποιούνται σύμφωνα με την ποιότητα και την απόστασή τους από την κυψέλη. Σε συγκεκριμένες στιγμές, μια αποικία μελισσών διαχωρίζεται: στη βασίλισσα και περίπου, στο ένα δεύτερο των εργαζόμενων που εγκαταλείπουν την κυψέλη, σε σμήνη, και διαμορφώνουν αρχικά μια ομάδα στο κλάδο ενός κοντινού δένδρου. Οι πιθανές περιοχές τοποθέτησής τους εξερευνούνται, προσεχτικά, με ανιχνεύσεις. Η επιλογή της περιοχής μπορεί να έχει διάρκεια αρκετών ημερών, κατά την διάρκεια των οποίων το σμήνος ρυθμίζει, ακριβώς, την θερμοκρασία του [Heinrich, B., 1981].

Η κατασκευή φωλιών από τις σφήκες Polybia Occidentalis περιλαμβάνει τρεις ομάδες εργαζομένων: προμηθευτές πολτού, νερού και οικοδόμους. Το μέγεθος της ομάδας είναι ρυθμισμένο σύμφωνα με τις ανάγκες της αποικίας μέσω κάποιας ροής πληροφοριών [Jeanne, R., 1996].

Μερικά είδη τερμιτών (Macrotermes) χτίζουν ακόμα και τις πιο σύνθετες φωλιές, που αποτελούνται από τους, κατά προσέγγιση, κωνικούς εξωτερικούς τοίχους (Σχήμα 3-1). Αυτοί αποτελούνται από τις ευδιάκριτες πλευρές που περιέχουν τους αγωγούς εξαερισμού, οι οποίοι αρχίζουν από την βάση του υποστηρίγματος μέχρι την κορυφή τους, από τις αίθουσες πληθυσμών μέσα στην κεντρική περιοχή «κυψελών», η οποία αποτελείται από λεπτά ελάσματα που υποστηρίζονται από τους στυλοβάτες, από ένα βασικό πιάτο με σπειροειδείς δροσερές διεξόδους. Αποτελούνται, επίσης, από μια βασιλική αίθουσα που είναι μια προστατευτική αποθήκη με μερικές μικρές τρύπες στους τοίχους της, μέσω των οποίων οι εργαζόμενοι μπορούν να περάσουν, από κήπους μυκήτων, που είναι ντυμένοι γύρω από την κυψέλη και περιλαμβάνουν ειδικές στοές ή κόμβους που βρίσκονται μεταξύ της εσωτερικής κυψέλης και των εξωτερικών τοίχων και τελικά από περιφερειακές στοές κατασκευασμένες πάνω και κάτω από το έδαφος, οι οποίες συνδέονται με τις περιοχές των προμηθευτικών κέντρων [Luscher, M., 1961].



Σχήμα 3-1: Τμήμα φωλιάς τερμιτών (Macrotermes Termites).
(1) Τοίχοι που περιέχουν αγωγούς εξαερισμού. (2) Αίθουσες πληθυσμών. (3) Βασικό πιάτο. (4) Βασιλική αίθουσα.

Κάθε έντομο εμφανίζει αρκετά σύνθετη δομή: χρησιμοποιώντας μεγάλο αριθμό αισθητήριων οργάνων συγκεντρώνει πληροφορίες από το περιβάλλον και παίρνει τις ανάλογες αποφάσεις. Ακόμα, η πολυπλοκότητα ενός μεμονωμένου εντόμου δεν είναι επαρκής για να εξηγηθεί η πολυπλοκότητα των αποφάσεων μιας αποικίας εντόμων.

3.2 Αυτο-οργάνωση

Πολλές πλευρές των συλλογικών δραστηριοτήτων των κοινωνικών εντόμων είναι αυτο-οργανωτικές. Η αυτο-οργάνωση είναι ένα φαινόμενο που περιγράφεται στους κλάδους της φυσικής και της βιολογίας. Ο Camazine (2000) προτείνει τον παρακάτω ορισμό για την αυτο-οργάνωση: *«Η αυτο-οργάνωση είναι μια διαδικασία μέσα στην οποία ένα μοντέλο (modele) ολικού επιπέδου αναδύεται (emergent) κατά μοναδικό τρόπο, από ένα μεγάλο αριθμό αλληλεπιδράσεων μεταξύ των συνιστωσών του χαμηλού επιπέδου του συστήματος. Οι κανόνες που χαρακτηρίζουν τις αλληλεπιδράσεις*

μεταξύ των συνιστωσών του συστήματος ακολουθούνται χρησιμοποιώντας μόνον τοπικές πληροφορίες χωρίς να λαμβάνουν υπόψη το συνολικό μοντέλο». Δυο όροι πρέπει να αποσαφηνιστούν: *μοντέλο* (modele) και *αναδυόμενη* ιδιότητα (emergent). Ο όρος *μοντέλο* είναι μια προσεγγιστική μετάφραση του αγγλικού όρου "pattern", που αφορά την έννοια της δομής και μπορεί, επίσης, να σημαίνει γενική διάταξη, σχήμα και τύπος. Μια *αναδυόμενη* ιδιότητα ενός συστήματος είναι, όσο την αφορά, ένα χαρακτηριστικό που εμφανίζεται αιφνιδίως (χωρίς να έχει προσδιοριστεί) από τις αλληλεπιδράσεις μεταξύ των συνιστωσών του συστήματος.

Μελετώντας την αυτο-οργάνωση των κοινωνικών εντόμων μιας αποικίας προκύπτουν χρήσιμα συμπεράσματα που αξιοποιούνται στο σχεδιασμό *ευφυών συστημάτων*. Στην πραγματικότητα, μια κοινωνική αποικία εντόμων είναι ένα αποκεντρωμένο σύστημα επίλυσης προβλημάτων, τα οποία αποτελούνται από πολλές απλές αλληλεπιδρούσες οντότητες. Τα καθημερινά προβλήματα που λύνονται μέσα σε μια αποικία περιλαμβάνουν την εύρεση των τροφίμων, την οικοδόμηση ή την επέκταση μιας φωλιάς, διαιρώντας αποτελεσματικά την εργασία μεταξύ των ατόμων, την παροχή τροφής στο πλήθος κ.λ.π. Πολλά από αυτά τα προβλήματα αντιστοιχούν σε ανάλογα που συναντώνται στη μηχανική και την πληροφορική. Ένα από τα πιο χαρακτηριστικά γνωρίσματα των κοινωνικών εντόμων είναι ότι λύνουν προβλήματα με μεγάλη *ευελιξία* (flexibility) και *στιβαρότητα* (robustness). Η ευελιξία επιτρέπει την προσαρμογή στις εκάστοτε περιβαλλοντικές συνθήκες, ενώ η στιβαρότητα «πριμοδοτεί» την αποικία με τη δυνατότητα να λειτουργήσει ακόμα και αν κάποια άτομα αποτύχουν.

Η αυτο-οργάνωση των κοινωνικών εντόμων μπορεί να ερμηνευθεί μέσω τεσσάρων μηχανισμών:

- Θετική ανάδραση (positive feedback),
- Αρνητική ανάδραση (negative feedback),
- Ενίσχυση των τυχαίων διακυμάνσεων (amplification of random fluctuations),
- Ύπαρξη αλληλεπιδράσεων μεταξύ των ατόμων της αποικίας.

Θετική ανάδραση (positive feedback)[1]

Θετική ανάδραση είναι μια διαδικασία με βάση την οποία τα κοινωνικά έντομα συγκλίνουν προς την κατεύθυνση καλών λύσεων ενός προβλήματος. Η θετική ανάδραση συμβάλει στην δημιουργία βασικών δομών. Περιλαμβάνει δυο μηχανισμούς, τη *στρατολόγηση* (recruitment) και την *ενίσχυση* (reinforcement).

Χαρακτηριστικό παράδειγμα στρατολόγησης είναι η περίπτωση ενός μυρμηγκιού που εντοπίζει το χώρο της τροφής του. Στη διάρκεια της επιστροφής του στη φωλιά του, εναποθέτει ποσότητες μιας χημικής ουσίας, (ονομάζεται φερομόνη), ανάλογες προς την ποσότητα και την ποιότητα της τροφής του. Οι ποσότητες της φερομόνης αποτελούν το *ίχνος* (trail) φερομόνης που ανιχνεύεται από τα υπόλοιπα μυρμήγκια της αποικίας, με αποτέλεσμα να δημιουργήσουν μεγάλη πιθανότητα να ακολουθήσουν την ίδια διαδρομή του μυρμηγκιού σε σχέση με άλλες τυχαίες διαδρομές. Στην περίπτωση αυτή το μυρμήγκι «στρατολογεί» τα άλλα μυρμήγκια με σκοπό να εκμεταλλευτούν την ίδια *πηγή τροφής* (food source).

Ο μηχανισμός της ενίσχυσης λειτουργεί στην περίπτωση που το μυρμήγκι εντοπίζει το ίχνος της φερομόνης και με κάποια πιθανότητα αποφασίζει να το ακολουθήσει [Fanekos, 2001]. Αν το ακολουθήσει τότε το ίχνος ενισχύεται, εναποθέτοντας και τη δική του φερομόνη. Με αυτό τον τρόπο η διαδρομή ενισχύεται με περισσότερη ποσότητα φερομόνης με αποτέλεσμα να γίνει πιο ελκυστική για τα υπόλοιπα μυρμήγκια.

Αρνητική ανάδραση (negative feedback)

Η *αρνητική ανάδραση* (negative feedback) αντισταθμίζει τη θετική ανάδραση και συμβάλει στη σταθεροποίηση κάποιας ορισμένης δομής. Ένα παράδειγμα αρνητικής ανάδρασης είναι η εξάτμιση της φερομόνης. Υπάρχουν διαδρομές των μυρμηγκιών που δεν έχουν αξία. Για αυτές τις διαδρομές προβλέπεται η εξάτμιση της εναποτιθέμενης φερομόνης.

[1] Η *θετική ανάδραση* είναι μια διαδικασία που αυτό- ενισχύεται, με ένα τρόπο που προκαλεί την πολύ γρήγορη σύγκλιση και, αν δεν υπάρχει κανένας μηχανισμός περιορισμού, οδηγεί στην έκρηξη.

Ενίσχυση των τυχαίων διακυμάνσεων (amplification of random fluctuations)

Η αυτο-οργάνωση στηρίζεται στην ενίσχυση των τυχαίων διακυμάνσεων (amplification of random fluctuations). Αυτή η σπουδαία διαδικασία οδηγεί στην δημιουργία νέων βελτιωμένων λύσεων ακόμα και της συνολικής βέλτιστης λύσης, ξεπερνώντας τα τοπικά βέλτιστα. Η τυχαιότητα αποτελεί κομβικό σημείο των πραγματικών βιολογικών συστημάτων. Για παράδειγμα, τα μυρμήγκια μπορούν να χαθούν κατά την αναζήτηση τροφής, αλλά τελικά να ανακαλύψουν νέες πηγές τροφών και να «καλέσουν» και άλλα μέλη της αποικίας τους για την εκμετάλλευση αυτών των πηγών τροφίμων.

Ύπαρξη αλληλεπιδράσεων

Όλα τα μοντέλα αυτο-οργάνωσης στηρίζονται στις *πολλαπλές αλληλεπιδράσεις* (multiple interactions). Κάθε άτομο – μυρμήγκι μπορεί να παράγει μια αυτο-οργανωμένη δομή, όπως ένα *μονοπάτι φερομόνης*[2]. Η αυτο-οργάνωση, γενικά, απαιτεί ένα ελάχιστο αριθμό αλληλεπιδρώντων ατόμων – μυρμηγκιών. Ακόμα τα άτομα – μυρμήγκια πρέπει να είναι σε θέση να χρησιμοποιήσουν τα αποτελέσματα των δραστηριοτήτων τους, όπως και άλλες δραστηριότητες. Για παράδειγμα, τα δίκτυα ιχνών φερομόνης μπορούν να αυτο-οργανωθούν και να χρησιμοποιηθούν συλλογικά αν τα άτομα χρησιμοποιούν τη φερομόνη των άλλων.

Τα αυτο-οργανωμένα συστήματα χαρακτηρίζονται από μερικές βασικές ιδιότητες [Bonabeau και άλλοι, 1999].

- Δημιουργία συγκεκριμένων δομών σε ένα, αρχικά, ομοιογενές μέσο. Τέτοιες δομές περιλαμβάνουν αρχιτεκτονικές φωλιών, ίχνη φερομόνης ή οργανωμένες κοινωνίες.

- Πιθανή συνύπαρξη διαφόρων σταθερών και αποδεκτών καταστάσεων (multistable). Επειδή οι δομές προκύπτουν από την ενίσχυση των τυχαίων αποκλίσεων, οποιαδήποτε τέτοια απόκλιση μπορεί να ενισχυθεί, και το σύστημα θα συγκλίνει σε μια μεταξύ των διαφόρων πιθανών καταστάσεων, ανάλογα με τις αρχικές συνθήκες. Για παράδειγμα, όταν παρουσιάζονται δυο ίδιες πηγές τροφής, στην ίδια απόσταση από τη φωλιά, σε μια αποικία μυρμηγκιών που προσφεύγει

[2] *Μονοπάτι φερομόνης* (pheromone trails) είναι κάθε διαδρομή στην οποία εναποτίθεται μια ποσότητα φερομόνης από ένα η περισσότερα μυρμήγκια.

στη μαζική στρατολόγηση, μία από αυτές αξιοποιείται, τελικά, μαζικά. Και οι δυο πηγές έχουν την ίδια πιθανότητα εκμετάλλευσης, αλλά μόνο μία από αυτές επιλέγεται. Υπάρχουν δυο πιθανές εκδοχές σε αυτό το παράδειγμα: ολική εκμετάλλευση της μιας ή ολική εκμετάλλευση της άλλης. Η τελική αξιοποίηση μιας εκ των δύο πηγών τροφής εξαρτάται από τις αρχικές τυχαίες διακυμάνσεις.

- Παρουσία διακλαδώσεων (bifurcation) όταν μερικές παράμετροι μεταβάλλονται. Η συμπεριφορά ενός αυτο-οργανωμένου συστήματος αλλάζει, εντυπωσιακά, στις διακλαδώσεις. Για παράδειγμα, ο τερμίτης *Macrotermes* χρησιμοποιεί σφαιρίδια χώματος εμποτισμένα με φερομόνη για να χτίσει στήλες. Δύο διαδοχικές φάσεις υλοποιούνται [Grasse, P-P, 1959]. Αρχικά, η μη συντονισμένη (non coordination) φάση χαρακτηρίζεται από μια τυχαία εναπόθεση σφαιριδίων χώματος. Αυτή η φάση διαρκεί μέχρι μια από τις εναποθέσεις σφαιριδίων χώματος φτάσει σε ένα κρίσιμο μέγεθος. Στη συνέχεια, η φάση συντονισμού (coordination) αρχίζει όταν η ομάδα «οικοδόμων» είναι αρκετά μεγάλη, με αποτέλεσμα να προκύψουν στήλες ή λουρίδες του παραπάνω υλικού. Η ύπαρξη μιας αρχικής εναπόθεσης σφαιριδίων χώματος υποκινεί τους εργάτες (workers)[3] για να συγκεντρώσουν περισσότερο υλικό μέσω του μηχανισμού της θετικής ανάδρασης, γιατί η συγκέντρωση του υλικού ενισχύει την ελαστικότητα των εναποθέσεων μέσω της φερομόνης που εκπέμπονται από τα σφαιρίδια χώματος [Bruinsma, O., H., 1979]. Αυτό το αυτο-καταλυτικό φαινόμενο οδηγεί σε μια συντονισμένη φάση. Αν ο αριθμός των «οικοδόμων» είναι πολύ μικρός, η φερομόνη εξατμίζεται μεταξύ δυο διαδοχικών μεταβάσεων από τους εργάτες, ενώ ο μηχανισμός της ενίσχυσης δεν λειτουργεί. Μόνο η μη συντονισμένη φάση παρατηρείται. Οπότε, δεν υπάρχει ανάγκη επίκλησης μιας αλλαγής της συμπεριφοράς από τους συμμετέχοντες στην μετάβαση από τη μη συντονισμένη, στη συντονισμένη φάση. Αυτό είναι αποτέλεσμα μιας αύξησης του μεγέθους της ομάδας.

3.3 Στιγμεργία και συντονισμός

Η *στιγμεργία* είναι μια μέθοδος επικοινωνίας σε αποκεντρωμένα συστήματα στα οποία τα άτομα – μέρη του συστήματος επικοινωνούν

[3] Οι εργάτες (workers) είναι μια κοινωνική τάξη των τερμιτών.

μεταξύ τους αλλάζοντας το τοπικό τους περιβάλλον. Γενικά, υπάρχουν δυο κύριες κατηγορίες συντονισμού μεταξύ των ατόμων – πρακτόρων :

- Συντονισμός με άμεση επικοινωνία, και
- Συντονισμός μέσω έμμεσης αλληλεπίδρασης.

Σε αυτή τη διατριβή χρησιμοποιείται ο συντονισμός μέσω έμμεσης αλληλεπίδρασης. Αξιοποιεί μια προσέγγιση εμπνευσμένη από τον τρόπο με τον οποίο οι αποικίες των μυρμηγκιών μεταδίδουν πληροφορίες κατά την αναζήτηση της τροφή τους [Theraulaz και άλλοι, 1999]. Ο όρος *στιγμεργία* (stigmergy) προέρχεται από τις ελληνικές λέξεις *στίγμα* (stigma) και *έργο* (ergo) και εισήχθη πρώτη φορά από τον Grasse (1959) στο έργο του για τους τερμίτες ***Bellicositetermes Natalensis*** και ***Cubitermes***. Στο έργο αυτό έχει ορίσει τον όρο στιγμεργία (stigmergy) ως εξής:

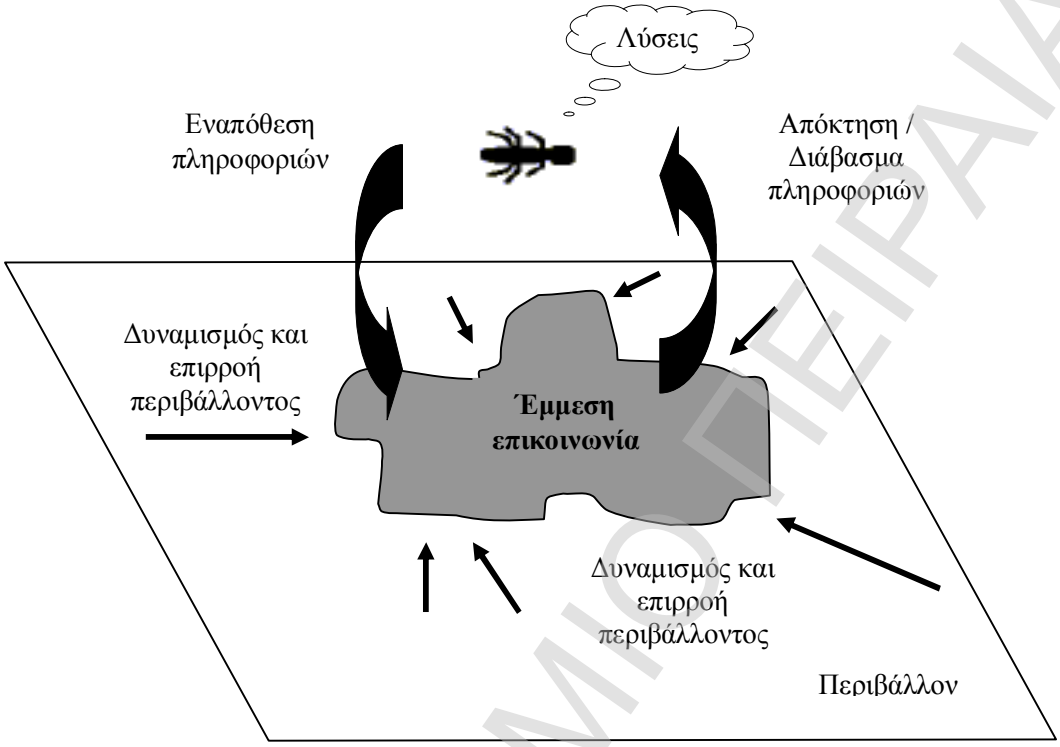
«Διέγερση των εργατών από την απόδοση που έχουν επιτύχει».

Η αλληλεπίδραση των μυρμηγκιών βασίζεται στην ύπαρξη μιας αρωματικής χημικής ουσίας, της *φερομόνης*. Τα μυρμήγκια εναποθέτουν φερομόνη στο περιβάλλον τους, η οποία παρατηρείται από τα αλλά μυρμήγκια και επηρεάζει την συμπεριφορά τους. Ο μηχανισμός της εναπόθεσης της φερομόνης αποτελείται από δυο κύριες διαδικασίες, εξαρτώμενες από το περιβάλλον στο οποίο τα έντομα βρίσκονται, και είναι η *συσσώρευση πληροφοριών* (aggregation) και η *εξάτμιση* της παλιάς εναποτιθέμενης φερομόνης (evaporation).

Ο μηχανισμός συντονισμού βασίζεται στην συμπεριφορά των μυρμηγκιών κατά την αναζήτηση τροφής. Κατά τη διάρκεια της αναζήτησης τροφής, τα μυρμήγκια ακολουθούν μια απλή διαδικασία κατά την οποία η συμπεριφορά τους επηρεάζεται από ένα μόνιμα μεταβαλλόμενο περιβάλλον. Τα μυρμήγκια αναζητούν την τροφή τους σύμφωνα με το παρακάτω τρόπο (Σχήμα 3-2):

- Όταν δεν υπάρχουν ίχνη φερομόνης στο περιβάλλον, τα μυρμήγκια πραγματοποιούν μια τυχαία αναζήτηση τροφής.
- Όταν ένα μυρμήγκι ανακαλύψει μια πηγή τροφής, εναποθέτει μια χημική αρωματική ουσία (φερομόνη) κατά την επιστροφή στην φωλιά και με αυτό το τρόπο δημιουργεί ένα μονοπάτι φερομόνης ανάμεσα

στη φωλιά και στην πηγή τροφής. Αυτή η φερομόνη θα εξατμιστεί αν κάποιο άλλο μυρμήγκι δεν εναποθέσει νέες ποσότητες φερομόνης.



Σχήμα 3-2: Διαδικασία λειτουργίας ενός μυρμηγκιού.

- Όταν ένα μυρμήγκι ανιχνεύσει τα ίχνη της φερομόνης, θα οδηγηθεί από το ένστικτό του και θα τα ακολουθήσει μέχρι την πηγή τροφής. Εντούτοις, υπάρχει μια μικρή πιθανότητα το μυρμήγκι να μην ακολουθήσει τα ίχνη φερομόνης. Όταν φτάσει στην πηγή τροφής, ενισχύει και ανανεώνει το μονοπάτι φερομόνης κατά την επιστροφή του στην φωλιά.

Κύριες ιδιότητες του παραπάνω μηχανισμού είναι οι εξής:

i. Η εξάτμιση της φερομόνης κάνει την αποικία να ξεχάσει τις πληροφορίες που δεν ισχύουν πια.

ii. Το περιβάλλον επαναχρησιμοποιείται (δεν υπάρχουν χάρτες στους εγκέφαλους των μυρμηγκιών). Αξιοσημείωτο είναι ότι η παρουσία του μονοπατιού φερομόνης αυξάνει την πιθανότητα τα μυρμήγκια να βρουν την πηγή τροφής, και να ξαναενισχύσουν το μονοπάτι κατά την επιστροφή τους πίσω στην φωλιά. Μια τέτοια ενέργεια αυτο-ενίσχυσης φαίνεται να είναι μια απαραίτητη ιδιότητα για

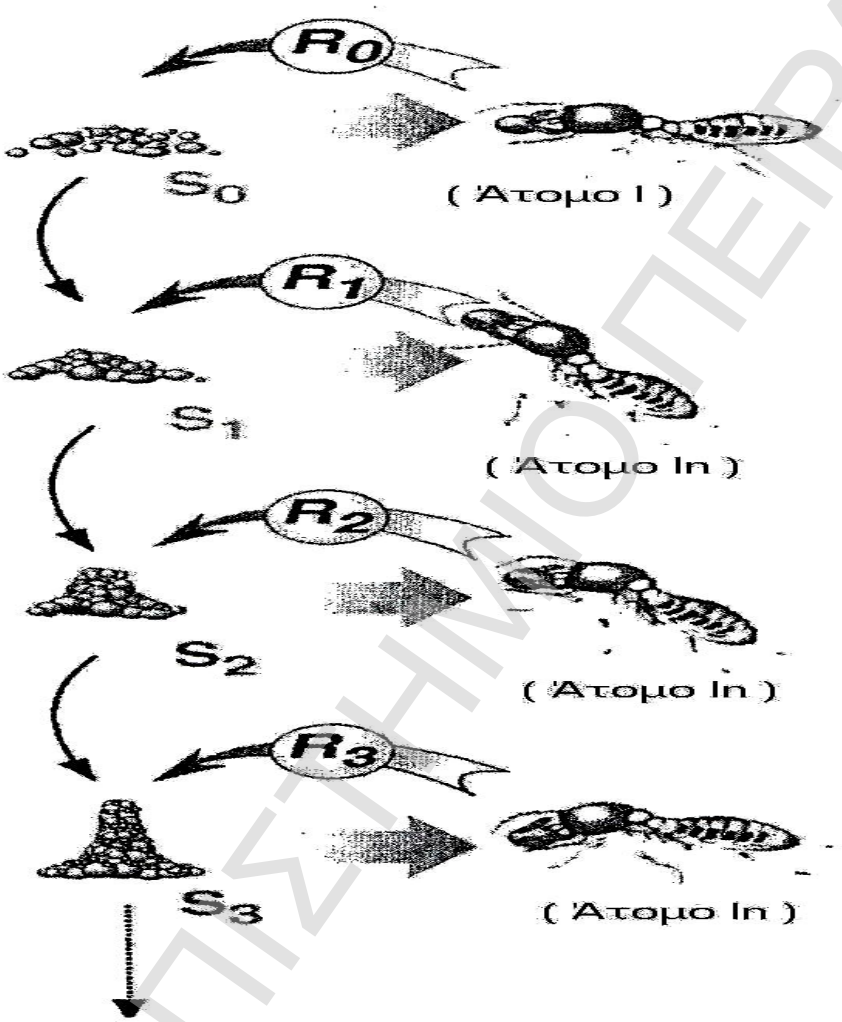
αποτελεσματική ανερχόμενη συμπεριφορά. Στο Σχήμα 3-2 περιγράφεται περιληπτικά πως ένα μυρμήγκι λειτουργεί κατά την αναζήτηση τροφής.

Υπενθυμίζεται ότι ο Grasse πρώτος εισήγαγε τον όρο στιγμεργία για να εξηγηθεί η συμπεριφορά της κοινωνίας των τερμιτών, ***Bellicositermes Natalensis*** και ***Cubitermes***, κατά τη διαδικασία δημιουργίας της φωλιάς τους. Αρχικά, οι τερμίτες εναποθέτουν τυχαία στο χώρο που πρόκειται να κτιστεί η φωλιά τους, χωμάτινα σφαιρίδια εμπλουτισμένα με τη φερομόνη τους [M. Dorigo και άλλοι, 2000]. Αυτή είναι η φάση μη συντονισμού. Στη συνέχεια, ο επόμενος τερμίτης ανιχνεύει τα σφαιρίδια και εναποθέτει το δικό του δίπλα τους (Σχήμα 3-3). Το αποτέλεσμα είναι να σχηματιστούν σωροί από χωμάτινα σφαιρίδια. Όταν κάποιος σωρός ξεπεράσει ένα ορισμένο μέγεθος, τότε δημιουργείται ένα νέο ερέθισμα που ενεργοποιεί τους τερμίτες να εναποθέσουν τα επόμενα σφαιρίδια, ούτως ώστε να σχηματιστεί μια στήλη. Αργότερα, σχηματίζονται τόξα που ενώνουν τις στήλες μεταξύ τους, οπότε και δημιουργείται η φωλιά τους.

Αν ο πληθυσμός των τερμιτών δεν είναι αρκετά μεγάλος, τότε λόγω της εξάτμισης της φερομόνης δεν συσσωρεύονται χωμάτινα σφαιρίδια, οπότε ο μηχανισμός ενίσχυσης δεν λειτουργεί με αποτέλεσμα να διαμορφώνεται μια μη συντονισμένη συμπεριφορά. Το παράδειγμα του Σχήματος 3-3 δείχνει ότι υπάρχει:

- Θετική ανάδραση (δημιουργία σωρών χωμάτινων σφαιριδίων και στηλών από σωρούς),
- Αρνητική ανάδραση (εξάτμιση φερομόνης),
- Ενίσχυση τυχαίων διακυμάνσεων (οι στήλες μπορούν να υλοποιηθούν οπουδήποτε),
- Έμμεση αλληλεπίδραση των ατόμων της αποικίας (στιγμεργία),
- Η ανάδειξη δομών (στήλες) μέσα σε ομοιογενές μέσο (το έδαφος με αρχικά τυχαία ομοιόμορφη κατανομή χωμάτινων σφαιριδίων),
- Η δυνατότητα πολλαπλών αποδεκτών λύσεων (εμφάνιση στηλών οπουδήποτε), και

- Μετάβαση από τη μια κατάσταση τυχαίας εναπόθεσης χωμάτινων σφαιριδίων σε άλλη κατάσταση που χαρακτηρίζεται από την οργανωμένη εναπόθεση χωμάτινων σφαιριδίων.



Σχήμα 3-3: Ένα παράδειγμα μιας διαδικασίας στιγμεργίας κατά την κατασκευή στηλών από τους τερμίτες. Αρχικά η αρχιτεκτονική βρίσκεται στην κατάσταση S_0, η οποία πυροδοτεί την αντίδραση R_0 του εργάτη I. Η S_0 αλλάζει από την δράση του I (για παράδειγμα, ο I μπορεί να ρίξει ένα χωμάτινο σφαιρίδιο) και μετασχηματίζεται σε μια νέα διεγερτική διαμόρφωση S_1 η οποία με την σειρά της μπορεί να πυροδοτήσει μια νέα αντίδραση R_1 από τον I ή οποιοδήποτε άλλον εργάτη I_n κ.ο.κ. Οι επιτυχημένες αντιδράσεις $R_1, R_2, \mathbf{K}, R_n$ μπορούν να παραχθούν από οποιοδήποτε εργάτη που μεταφέρει χωμάτινα

σφαιρίδια. Κάθε εργάτης προκαλεί ένα νέο ερέθισμα σε αντιδράσεις των υπαρχόντων διεγερμένων διαμορφώσεων. Αυτό το νέο ερέθισμα δρα στον ίδιο τερμίτη ή σε κάποιον άλλο εργάτη της αποικίας. Μια τέτοια διαδικασία, όπου μόνο οι σχετικές αλληλεπιδράσεις που λαμβάνουν χώρα ανάμεσα στους τερμίτες είναι έμμεσες, μέσω του περιβάλλοντος το οποίο αλλάζει από τους άλλους τερμίτες, είναι επίσης γνωστή κι ως ημιτεκτονική επικοινωνία [Wilson, E. O., 1975].

Η στιγμεργία συνδέεται, συχνά, με τους όρους ευελιξία και προσαρμοστικότητα.. Όταν το περιβάλλον αλλάζει λόγω μιας εξωτερικής διαταραχής, τα άτομα της αποικίας αποκρίνονται κατάλληλα σε εκείνη την διαταραχή, σαν να ήταν μια τροποποίηση του περιβάλλοντος που προκλήθηκε από τις δραστηριότητες της αποικίας. Η αποικία πρέπει να αποκριθεί *συλλογικά* στη διαταραχή, επιδεικνύοντας τα άτομα την ίδια συμπεριφορά. Οι τεχνητοί πράκτορες μπορούν να αποκριθούν σε μια διαταραχή χωρίς επαναπρογραμματισμό, για να εξετάσουν εκείνη την ιδιαίτερη διαταραχή.

3.4 Συμπεριφορά των μυρμηγκιών κατά τη συλλογή της τροφής τους

Σε αυτή την παράγραφο μελετάται η συμπεριφορά των μυρμηγκιών κατά την συλλογή της τροφής τους και ένα στοχαστικό μοντέλο, μέσω των εξισώσεων διαφορών, που εμπνέεται από αυτή τους την συμπεριφορά, στην οποία βασίζονται οι αλγόριθμοι βελτιστοποίησης αποικιών μυρμηγκιών [Foundas και Vlachos, 2005].

Τα μυρμήγκια, όταν κινούνται από την φωλιά τους προς μια πηγή τροφίμων ή όταν επιστρέφουν προς τη φωλιά τους ή και τα δύο, ανάλογα το είδος των μυρμηγκιών, δημιουργούν μονοπάτια φερομόνης (pheromone trails) τα οποία ακολουθούν τα υπόλοιπα μυρμήγκια [Goss, S. και άλλοι, 1989].

Η διαδικασία κατά την οποία τα μυρμήγκια επηρεάζονται από κάποια άλλα για να εκμεταλλευτούν μια πηγή τροφής ονομάζεται *στρατολόγηση* (recruitment). Όταν η στρατολόγηση βασίζεται μόνο στη φερομόνη ονομάζεται *μαζική στρατολόγηση* (mass recruitment).

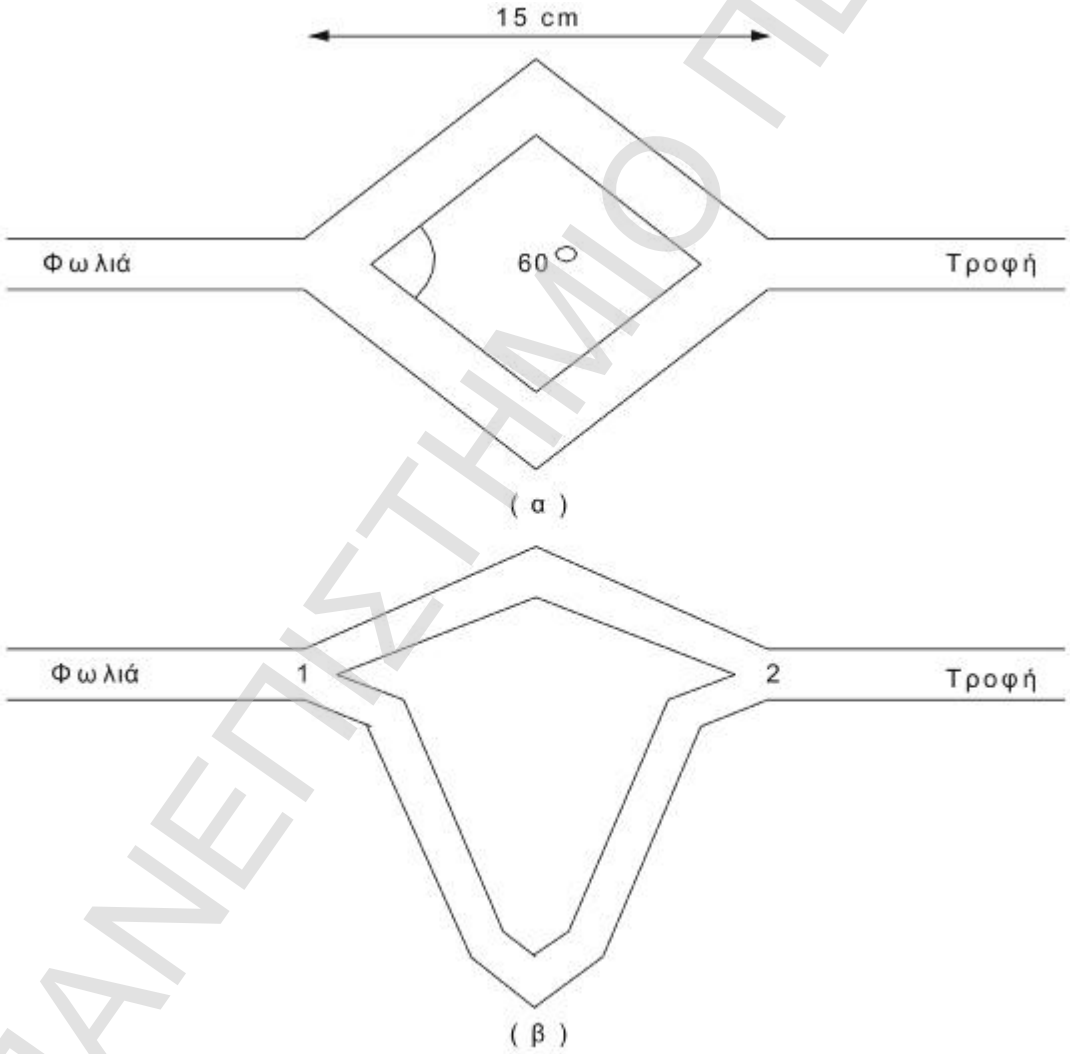
3.4-1 Πειραματική διάταξη γέφυρας δύο κλάδων

Η συμπεριφορά μερικών ειδών μυρμηγκιών, για παράδειγμα των *I. Humilis* [Goss και άλλοι, 1989] και των *Linepithema humile*, και *Lasius* [Bonabeau και άλλοι, 1997] βασίζεται στην έμμεση επικοινωνία τους μέσω της φερομόνης. Ενώ μετακινούνται από την πηγή τροφής προς τη φωλιά τους και αντίστροφα, τα μυρμήγκια εναποθέτουν στη γη χημικές ουσίες – φερομόνες – με αποτέλεσμα να δημιουργείται ένα μονοπάτι φερομόνης. Το μονοπάτι φερομόνης λειτουργεί ελκυστικά για τα υπόλοιπα μυρμήγκια της αποικίας τα οποία τείνουν να επιλέξουν, με βάση ένα πιθανοκρατικό νόμο, διαδρομές στις οποίες οι συγκεντρώσεις φερομόνης είναι μεγάλη.

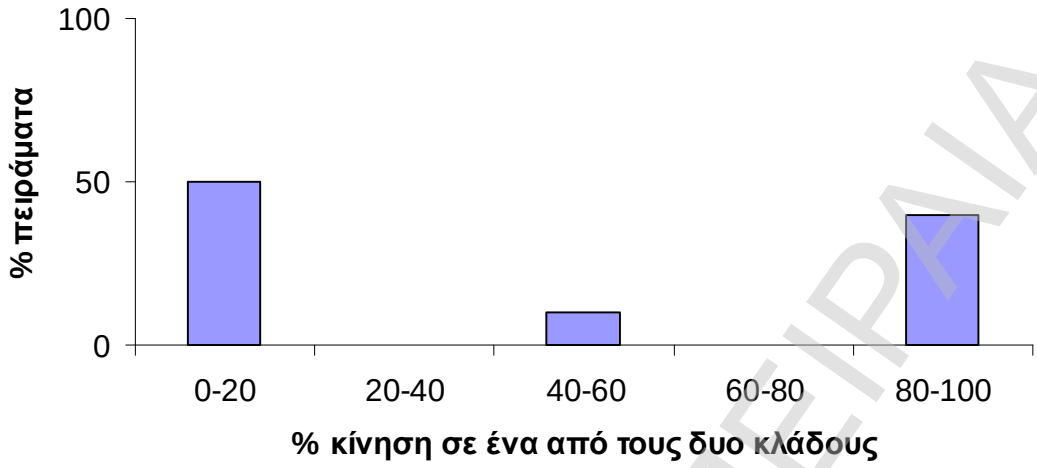
Η εναπόθεση της φερομόνης και η συμπεριφορά μερικών ειδών μυρμηγκιών ερευνήθηκε σε βάθος από ένα ορισμένο αριθμό ερευνητών, σε ειδικές πειραματικές διατάξεις. Ο Deneubourg και οι συνεργάτες του (1990), ο Goss και οι συνεργάτες του (1989) πραγματοποίησαν μια σειρά πειραμάτων σε διάταξη γέφυρας με δύο κλάδους, χρησιμοποιώντας μυρμήγκια I. Humilis από την Αργεντινή. Τα πειράματα υλοποιήθηκαν μεταβάλλοντας το λόγο $r = \frac{l_1}{l_s}$ του μήκους των δύο κλάδων της γέφυρας, όπου l_1 είναι το μήκος του μεγάλου κλάδου και l_s είναι το μήκος του μικρού κλάδου.

Στο πρώτο πείραμα της γέφυρας τα μήκη των δύο κλάδων είναι ίσα ($r = 1$, Σχήμα 3-4(α)). Αρχικά, τα μυρμήγκια αφέθηκαν ελεύθερα να μετακινηθούν ανάμεσα στην φωλιά και την πηγή τροφής και παρατηρήθηκε το ποσοστό των μυρμηγκιών που επέλεξαν το έναν ή τον άλλο από τους δυο κλάδους. Το αποτέλεσμα (Σχήμα 3-5(α)) ήταν ότι, αν και στην αρχική φάση πραγματοποιήθηκαν τυχαίες επιλογές, τελικά, όλα τα μυρμήγκια χρησιμοποίησαν το ίδιο κλάδο. Αυτό το αποτέλεσμα εξηγείται ως εξής: όταν μια δοκιμή αρχίζει δεν υπάρχει καθόλου φερομόνη στους δυο κλάδους. Εξαιτίας αυτού, τα μυρμήγκια δεν έχουν κάποια προτίμηση και διαλέγουν με την ίδια πιθανότητα οποιοδήποτε κλάδο. Στη συνέχεια, εξαιτίας τυχαίων διακυμάνσεων, μερικά μυρμήγκια θα προτιμήσουν έναν από τους δυο κλάδους. Επειδή τα μυρμήγκια εναποθέτουν φερομόνη καθώς περπατούν, ένας μεγαλύτερος αριθμός μυρμηγκιών σε έναν κλάδο έχει ως αποτέλεσμα ένα μεγαλύτερο ποσό φερομόνης σε αυτό τον κλάδο. Με την σειρά του αυτό το μεγαλύτερο ποσό παρακινεί περισσότερα μυρμήγκια να επιλέξουν αυτό το κλάδο ξανά, κοκ , μέχρι τελικά τα μυρμήγκια να συγκλίνουν σε ένα μοναδικό κλάδο. Αυτή η διαδικασία, αυτο-καταλυτικής ή θετικής ανάδρασης, είναι στην πραγματικότητα ένα παράδειγμα αυτο-οργανωτικής

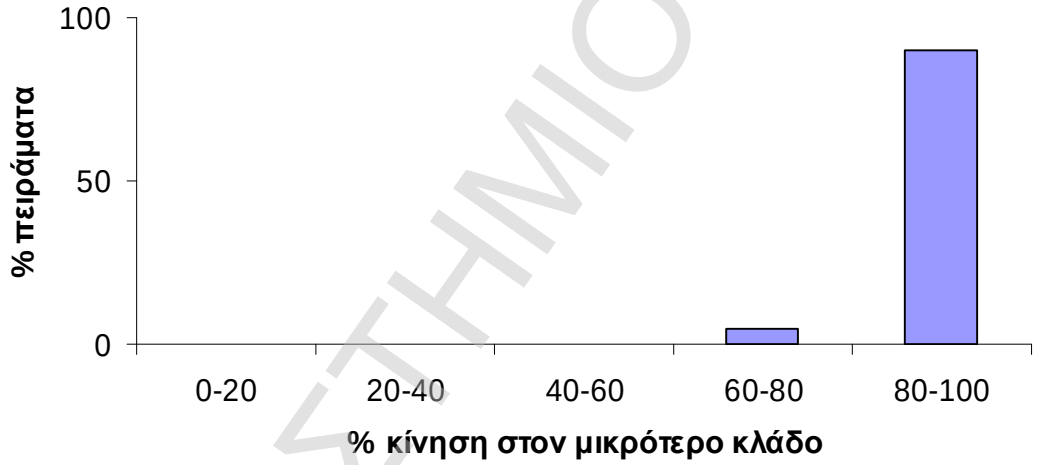
συμπεριφοράς των μυρμηγκιών: ένα μακροσκοπικό πρότυπο (αντίστοιχο της σύγκλισης σε ένα κλάδο) αναδύεται μέσα από διαδικασίες και αλληλεπιδράσεις, οι οποίες λαμβάνουν χώρα σε ένα «μικροσκοπικό» επίπεδο [Camazine και άλλοι, 2001]. Στην περίπτωσή μας, η σύγκλιση των μυρμηγκιών σε ένα μόνο κλάδο αντιπροσωπεύει τη μακροσκοπική συγκεντρωτική συμπεριφορά, η οποία εξηγείται από τη μικροσκοπική δραστηριότητα των μυρμηγκιών, δηλαδή από τις τοπικές αλληλεπιδράσεις μεταξύ των ατόμων της αποικίας. Είναι επίσης παράδειγμα στιγμεργικής επικοινωνίας. Τα μυρμήγκια συντονίζουν τις δραστηριότητες τους, χρησιμοποιώντας έμμεση επικοινωνία η οποία παρατηρείται από τροποποιήσεις του περιβάλλοντος στο οποίο κινούνται.



Σχήμα 3-4: Πειραματική διάταξη γέφυρας με δυο κλάδους. (α) Οι κλάδοι έχουν ίσο μήκος. (β) Οι κλάδοι έχουν διαφορετικό μήκος [Goss, 1989].



(α)



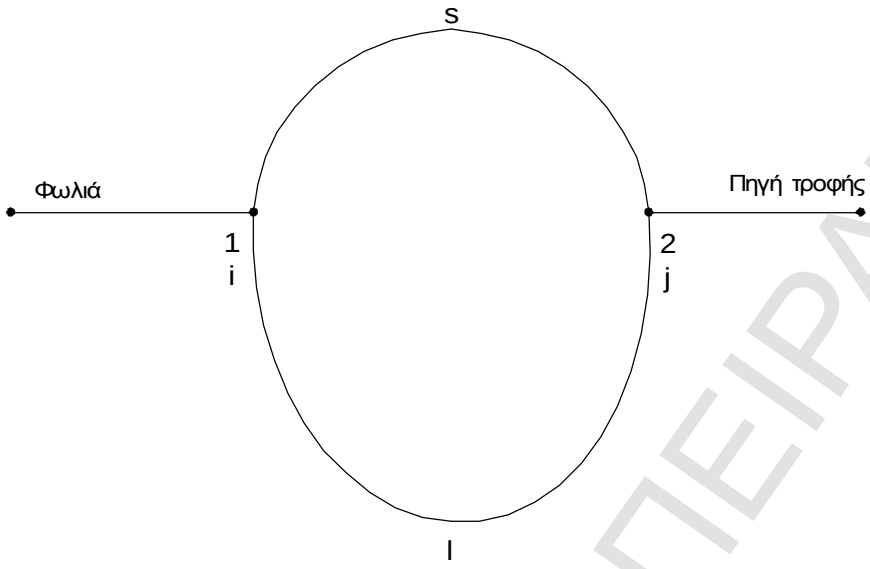
(β)

Σχήμα 3-5: Αποτελέσματα για τα *Iridonmyrmex humilies* μυρμήγκια από το πείραμα της γέφυρας με δύο κλάδους. (α) Αποτελέσματα που στην περίπτωση που οι δύο κλάδοι έχουν το ίδιο μήκος ($r=1$). Σε αυτή την περίπτωση τα μυρμήγκια χρησιμοποιούν είτε τον έναν κλάδο ή τον άλλο σε περίπου τον ίδιο αριθμό δοκιμών.
(β) Αποτελέσματα στην περίπτωση που ο ένας κλάδος έχει διπλάσιο μήκος από τον άλλο ($r=2$). Σε αυτή την περίπτωση σε όλες τις δοκιμές η μεγαλύτερη πλειοψηφία των μυρμηγκιών επιλέγουν τον μικρότερο κλάδο [Goss, 1989].

Στο δεύτερο πείραμα, η αναλογία μήκους μεταξύ των δύο κλάδων τέθηκε $r = 2$ [Goss, 1989], έτσι ώστε ο μεγαλύτερος κλάδος να είναι ο διπλάσιος του μικρότερου (Στο Σχήμα 3-4(β) δείχνεται η πειραματική διάταξη). Σε αυτή την περίπτωση, στις περισσότερες των προσπαθειών, μετά από ορισμένο χρόνο τα μυρμήγκια επιλέγουν να χρησιμοποιήσουν τον μικρότερο κλάδο (Σχήμα 3-5(β)). Όπως και στο πρώτο πείραμα, τα μυρμήγκια αφήνουν την φωλιά τους για να εξερευνήσουν το περιβάλλον και φθάνουν σε ένα σημείο απόφασης όπου πρέπει να διαλέξουν έναν από τους δύο κλάδους. Επειδή οι δύο κλάδοι αρχικά μοιάζουν, τα μυρμήγκια διαλέγουν τυχαία. Επομένως, το αναμενόμενο είναι μισά από τα μυρμήγκια να διαλέξουν τον μικρό κλάδο και τα αλλά μισά τον μεγάλο, παρόλο που στοχαστικές ταλαντώσεις μπορεί περιστασιακά να ευνοούν τον έναν από τους δύο κλάδους. Εν τούτοις, αυτή η πειραματική διάταξη παριστάνει μια αξιοθαύμαστη διαφορά σε σχέση με την προηγούμενη. Επειδή ο ένας κλάδος είναι μικρότερος από τον άλλον (Σχήμα 3-4(β)), τα μυρμήγκια που επιλέγουν τον μικρότερο κλάδο φθάνουν πρώτα στην τροφή και έτσι ξεκινούν την επιστροφή τους στην φωλιά. Αλλά τότε, όταν πρέπει να αποφασίσουν μεταξύ του μεγάλου και του μικρού κλάδου, το υψηλό επίπεδο φερομόνης στον μικρό κλάδο θα τα οδηγήσει να διαλέξουν αυτόν. Με αυτό τον τρόπο, η φερομόνη αρχίζει να συγκεντρώνεται πιο γρήγορα στον μικρό κλάδο, ο οποίος τελικά θα χρησιμοποιηθεί από όλα τα μυρμήγκια εξαιτίας της αυτο-καταλυτικής διαδικασίας που περιγράφηκε προηγουμένως. Συγκριτικά με το πείραμα με τους δύο ισομήκεις κλάδους, η επιρροή των αρχικών τυχαίων διακυμάνσεων μειώνεται πολύ και η στιγμεργία, η αυτο-κατάλυση και το διαφορικό μήκος του μονοπατιού είναι αυτά που παίζουν κύριο ρόλο. Με ενδιαφέρον, παρατηρείται επίσης ότι, ακόμα και όταν ο μεγάλος κλάδος είναι διπλάσιος του μικρού, δεν χρησιμοποιούν όλα τα μυρμήγκια τον μικρό κλάδο αλλά ένα μικρό ποσοστό μπορεί να πάει από τον μεγάλο. Αυτό μπορεί να ερμηνευτεί ως ένας τύπος «εκμετάλλευσης μονοπατιού».

3.4-2 Ένα στοχαστικό μοντέλο μέσω Εξισώσεων Διαφορών

Ο Deneubourg και οι συνεργάτες του (1990) πρότειναν ένα στοχαστικό μοντέλο που περιγράφει την δυναμική της κοινωνίας των μυρμηγκιών όπως παρατηρείται στο πείραμα της γέφυρας με δύο κλάδους. Στο Σχήμα (3-6), δείχνεται η πειραματική διάταξη γέφυρας με κλάδους διαφορετικού μήκους. Έστω Ψ μυρμήγκια διανύουν ανά δευτερόλεπτο τη γέφυρα προς κάθε κατεύθυνση με σταθερή ταχύτητα υ cm/s και εναποθέτουν μια μονάδα φερομόνης σε κάθε κλάδο.



Σχήμα 3-6: Πειραματική διάταξη γέφυρας με κλάδους διαφορετικού μήκους. Τα μυρμήγκια κινούνται από τη φωλιά τους προς την πηγή τροφή τους και αντίστροφα.

Αν l_s και l_1 είναι τα μήκη, σε cm, του μεγαλύτερου (l) και του μικρότερου (s) κλάδου της γέφυρας, ένα μυρμήγκι επιλέγει να διανύσει το μικρότερο (s) κλάδο σε χρόνο $t_s = \frac{l_s}{\upsilon}$ δευτερόλεπτα, ενώ ο χρόνος (t_1)που απαιτείται για να διανύσει ένα μυρμήγκι το μεγαλύτερο (l) κλάδο είναι $t_1 = rt_s$ δευτερόλεπτα, με $r = \frac{l_1}{l_s}$

<u>Πρόταση 3.1</u>: Αν ο χρόνος που απαιτείται, t_1, να διανύσει ένα μυρμήγκι, το μεγαλύτερο κλάδο, l_1, με σταθερή ταχύτητα υ cm/s, είναι $t_1 = \frac{l_1}{\upsilon}$, τότε $t_1 = rt_s$.

<u>Απόδειξη</u>: Πράγματι είναι $t_1 = \frac{l_1}{\upsilon}$ οπότε $t_1 = \frac{l_1}{l_s}\frac{l_s}{\upsilon}$ δηλαδή $t_1 = rt_s$ όπου $r = \frac{l_1}{l_s}$ με $r > 1$.

Η πιθανότητα $P_{i\alpha}(t)$ για ένα μυρμήγκι να φθάσει στο σημείο απόφασης $i \in \{1,2\}$ και να επιλέξει τον κλάδο $\alpha \in \{s,1\}$ δίνεται [Dorigo και Stützle, 2004]:

$$P_{i\alpha} = \frac{(t_s + \Phi_{is}(t))^{\alpha}}{(t_s + \Phi_{is}(t))^{\alpha} + (t_s + \Phi_{il}(t))^{\alpha}} \tag{3.1}$$

όπου $\Phi_{i\alpha}(t)$ είναι η συνολική ποσότητα της φερομόνης στον κλάδο της γέφυρας, η οποία είναι ανάλογη του αριθμού των μυρμηγκιών που χρησιμοποιούν τον κλάδο στον χρόνο t. Πειραματικά προκύπτει ότι $\alpha = 2$. Ισχύει $P_{is}(t) + P_{il}(t) = 1$. Από το δυναμικό σύστημα [Dorigo και Stützle, 2004]:

$$\frac{d\Phi_{is}}{dt} = \Psi P_{is}(t - t_s) + \Psi P_{is}(t), \quad i, j \in \{1,2\}, i \neq j \tag{3.2}$$

και

$$\frac{d\Phi_{il}}{dt} = \Psi P_{il}(t - rt_s) + \Psi P_{il}(t) \quad i, j \in \{1,2\}, i \neq j \tag{3.3}$$

προκύπτει η παρακάτω πρόταση [Foundas and Vlachos, 2005].

<u>*Πρόταση 3.2*</u>: Η ποσότητα της φερομόνης $\Phi_{is}(t)$ (αντίστοιχα $\Phi_{il}(t)$) που εναποτίθεται στους κλάδους s (αντίστοιχα l) δίνεται από τη λύση του επόμενου συστήματος εξισώσεων διαφορών:

$$\Phi_{is}(t+1) - \Phi_{is}(t) = \Psi\left[P_{js}(t - t_s) + P_{is}(t)\right] \tag{3.4}$$

$$\Phi_{il}(t+1) - \Phi_{il}(t) = \Psi\left[P_{jl}(t - t_l) + P_{il}(t)\right] \tag{3.5}$$

<u>*Απόδειξη*</u>: Η εξίσωση διαφορών (3.4) (όπως αντίστοιχα και η (3.5)) είναι πρώτου βαθμού μη ομογενής. Έστω $\Phi_{is}(t+1) \equiv y_{t+1}$, $\Phi_{is}(t) \equiv y_t$ και $\Psi\left[P_{js}(t - t_s) + P_{is}(t)\right] \equiv g_t$, οπότε η εξίσωση (4) γίνεται:

$$y_{t+1} - y_t = g_t \tag{3.6}$$

Η αντίστοιχη ομογενής εξίσωση διαφορών είναι:

$$y_{t+1} - y_t = 0 \tag{3.7}$$

η οποία αλληλοδιαδοχικά γράφεται:

$$\begin{aligned} y_2 &= y_1 \\ y_3 &= y_2 \\ &\mathbb{M} \quad\quad \mathbb{M} \\ y_t &= y_{t-1} \end{aligned} \tag{3.8}$$

Πολλαπλασιάζοντας τις (3.8) κατά μέλη, προκύπτει:

$$y_t = y_1 = k = \text{σταθερά} \tag{3.9}$$

Η εξίσωση διαφορών (3.6) γράφεται:

$\Delta^1(y_t) = g_t$ οπότε

$y_t = \Delta^{-1}(g_t)$ και

$$y_t = \sum_{\lambda=1}^{t-1} g_\lambda \tag{3.10}$$

Η γενική λύση είναι:

$$y_t = k + \sum_{\lambda=1}^{t-1} g_\lambda$$

ή $y_t = k + y_t = k + \sum_{\lambda=1}^{t-1} \Psi\left[P_{is}(t - t_s) + P_{is}(t)\right]$

$$\text{ή } \Phi_{is}(t) = k + \sum_{\lambda=1}^{t-1} \Psi\left[P_{js}(t - t_s) + P_{is}(t)\right] \tag{3.11}$$

Με τον ίδιο τρόπο η λύση της εξίσωσης (3.5) είναι:

$$\Phi_{il}(t) = k + \sum_{\lambda=1}^{t-1} \Psi\left[P_{jl}(t - t_l) + P_{il}(t)\right] \tag{3.12}$$

Η εξίσωση (3.11) εκφράζει τη μεταβολή της φερομόνης, στο χρόνο t, του κλάδου s, και στο σημείο απόφασης i δίνεται από την ποσότητα των μυρμηγκιών Ψ, που θεωρείται σταθερή, πολλαπλασιασμένη με την πιθανότητα επιλογής του μικρότερου κλάδου στο σημείο απόφασης j στο χρόνο $t - t_s$ και με την πιθανότητα επιλογής του μικρότερου κλάδου στο σημείο απόφασης 1 στο χρόνο t, προστιθέμενης της σταθεράς k. Η σταθερά t_s είναι ο χρόνος που απαιτείται για να διανύσουν τα μυρμήγκια το συντομότερο κλάδο. Η εξίσωση (3.12) αφορά την μεταβολή της φερομόνης στο μεγαλύτερο κλάδο, όταν ο χρόνος που απαιτείται για να τον διανύσουν τα μυρμήγκια είναι $t - t_l$.

3.5 Από τα πραγματικά στα τεχνητά μυρμήγκια

Στους μετα-ευρεστικούς αλγορίθμους βελτιστοποίησης αποικιών μυρμηγκιών (ACO) μια αποικία *τεχνητών μυρμηγκιών*[4] εργάζεται συλλογικά για την εύρεση καλών λύσεων σε δύσκολα διακριτά προβλήματα βελτιστοποίησης. Η συλλογική εργασία αποτελεί το κλειδί για το σχεδιασμό των συνιστωσών – μερών των αλγορίθμων ACO. Στην πράξη πρέπει να συνδεθούν τα κατάλληλα υπολογιστικά εργαλεία με ένα σύνολο τεχνητών μυρμηγκιών τα οποία επικοινωνούν έμμεσα, με στιγμεργικό τρόπο. Οι καλές λύσεις αποτελούν μια αναδυόμενη (emergent) ιδιότητα της αλληλεπίδρασης των συνεργαζόμενων ατόμων – πρακτόρων.

Τα τεχνητά μυρμήγκια έχουν μια διπλή φύση [Dorigo και άλλοι, 1999]:

i. αποτελούν προσομοίωση, των συμπεριφορών, των πραγματικών μυρμηγκιών, και

ii. εμπλουτίζονται με ικανότητες που δεν απαντώνται στα πραγματικά μυρμήγκια. Με αυτό το τρόπο αυξάνεται η απόδοση και η αποτελεσματικότητα των αλγορίθμων ACO.

Υπάρχουν ομοιότητες μεταξύ των τεχνητών και πραγματικών μυρμηγκιών [Dorigo και άλλοι, 1999]. Ειδικότερα, στα τεχνητά μυρμήγκια χρησιμοποιείται:

[4] Ο όρος *artificial ants* αντιστοιχεί στον όρο τεχνητά μυρμήγκια. Στη συνέχεια τα τεχνητά μυρμήγκια θα αναφέρονται με τον όρο μυρμήγκια ή άτομα ή πράκτορες.

i. μια αποικία συνεργαζόμενων ατόμων,

ii. ένα τεχνητό ίχνος φερομόνης για τοπική στιγμεργική επικοινωνία,

iii. μια ακολουθία τοπικών κινήσεων για την εύρεση των συντομότερων διαδρομών, και

iv. μια στοχαστική πολιτική αποφάσεων, αξιοποιώντας τοπικές πληροφορίες.

Αποικία συνεργαζόμενων ατόμων

Όπως στις αποικίες των πραγματικών μυρμηγκιών, έτσι και στους αλγορίθμους των μυρμηγκιών, ένας πληθυσμός ή μια αποικία με την συλλογική συνεργασία τους συντελούν να βρεθεί μια «καλή λύση», ενός συγκεκριμένου προβλήματος. Αν και η πολυπλοκότητα κάθε τεχνητού μυρμηγκιού είναι τέτοια που μπορεί να υλοποιήσει μια εφικτή λύση (δεδομένου ότι ένα πραγματικό μυρμήγκι μπορεί να βρει, με κάποιο τρόπο, μια διαδρομή μεταξύ φωλιάς και πηγή τροφής) οι υψηλές ποιοτικές λύσεις είναι το αποτέλεσμα της συνεργασίας των ατόμων, ολόκληρης, της αποικίας. Τα μυρμήγκια συνεργάζονται με τη βοήθεια πληροφοριών που διαβάζονται ταυτόχρονα και καταγράφονται στις θέσεις (states) του προβλήματος που επισκέπτονται. Ο αριθμός των μυρμηγκιών σχετίζεται με τον τύπο του προβλήματος.

Ίχνος φερομόνης και στιγμεργία

Τα τεχνητά μυρμήγκια τροποποιούν μερικές πλευρές του περιβάλλοντος που κινούνται, όπως ακριβώς, τα πραγματικά μυρμήγκια. Ενώ τα πραγματικά μυρμήγκια εναποθέτουν στη γη μια χημική ουσία, τη φερομόνη, τα τεχνητά μυρμήγκια αλλάζουν μερικές αριθμητικές πληροφορίες που αποθηκεύονται τοπικά στις θέσεις του προβλήματος που επισκέπτονται. Αυτές οι αριθμητικές πληροφορίες παίρνουν την μορφή του *τεχνητού ίχνους φερομόνης* ή ίχνος φερομόνης για συντομία.

Στους αλγορίθμους ACO, τα τοπικά ίχνη φερομόνης είναι τα μοναδικά κανάλια επικοινωνίας μεταξύ των μυρμηγκιών. Αυτή η στιγμεργική μορφή επικοινωνίας διαδραματίζει σημαντικό ρόλο στη χρησιμοποίηση της συλλογικής γνώσης. Συνήθως, στους αλγορίθμους ACO ένας μηχανισμός *εξάτμισης*, παρόμοιος με την πραγματική εξάτμιση

φερομόνης, τροποποιεί τις πληροφορίες των φερομονών στη διάρκεια του χρόνου. Η εξάτμιση φερομονών επιτρέπει στην αποικία των μυρμηγκιών να ξεχάσει αργά την προηγούμενη ιστορία της, έτσι ώστε να μπορεί να κατευθύνει την αναζήτησή της, προς νέες κατευθύνσεις χωρίς να αυτοπεριορίζεται από προηγούμενες αποφάσεις. Η εξάτμιση νομιμοποιεί την ύπαρξη του μηχανισμού της αρνητικής ανάδρασης και συντελεί στην αποφυγή του εγκλωβισμού των ατόμων σε τοπικά βέλτιστα.

Αναζήτηση της συντομότερης διαδρομής

Τα τεχνητά και τα πραγματικά μυρμήγκια έχουν ένα κοινό στόχο: να βρουν την συντομότερη διαδρομή (ελάχιστο κόστος) συνδέοντας μια αρχή (φωλιά) με τις περιοχές προορισμού τους (πηγή τροφής). Τα πραγματικά μυρμήγκια διασχίζουν τις γειτονικές θέσεις των περιοχών της γης, όπως και τα τεχνητά κινούνται βαθμιαία μέσω των γειτονικών θέσεων του προβλήματος.

Πιθανοκρατικός τρόπος μετάβασης

Τα τεχνητά μυρμήγκια, όπως και τα πραγματικά, δημιουργούν λύσεις εφαρμόζοντας ένα πιθανοκρατικό νόμο αποφάσεων στις γειτονιές των θέσεων ενός προβλήματος. Όπως στα πραγματικά μυρμήγκια, η πολιτική αποφάσεων των τεχνητών μυρμηγκιών χρησιμοποιεί μόνο τις τοπικές πληροφορίες, χωρίς να την ενδιαφέρει η μελλοντική πρόβλεψη των γειτονιών του προβλήματος. Οπότε η πολιτική αποφάσεων που εφαρμόζεται έχει, απόλυτα, τοπικό χαρακτήρα στο χώρο και στο χρόνο. Οι τοπικές πληροφορίες κατατάσσονται σε δυο κατηγορίες:

i. πληροφορίες που είναι γνωστές εκ των προτέρων (a priori) και εκφράζονται μέσω της *ευρεστικής πληροφορίας* (heuristic information), και

ii. πληροφορίες που προέρχονται από τις τοπικές αλλαγές του περιβάλλοντος (ίχνη φερομόνης).

Τα τεχνητά μυρμήγκια έχουν μερικά χαρακτηριστικά που δεν τα έχουν τα πραγματικά μυρμήγκια:

- Τα τεχνητά μυρμήγκια ζουν σε *διακριτό* κόσμο και οι κινήσεις τους συνίστανται σε μεταβάσεις από διακριτές σε διακριτές θέσεις.

- Τα τεχνητά μυρμήγκια έχουν μια *εσωτερική θέση* (internal state). Αυτή η ιδιωτική θέση αποτελεί την μνήμη του μυρμηγκιού για προηγούμενες δράσεις του.

- Τα τεχνητά μυρμήγκια εναποθέτουν μια ποσότητα φερομόνης η οποία είναι συνάρτηση της *ποιότητας λύσης* του προβλήματος.

- Ο τεχνητός συγχρονισμός των μυρμηγκιών στην εναπόθεση της φερομόνης είναι ένα εξαρτώμενο πρόβλημα και δεν αντανακλά τη συμπεριφορά των πραγματικών μυρμηγκιών. Για παράδειγμα, σε μερικές περιπτώσεις, τα τεχνητά μυρμήγκια ενημερώνουν τα ίχνη φερομόνης μόνο μετά την παραγωγή μιας λύσης.

- Για να βελτιωθεί η αποδοτικότητα των αλγορίθμων ACO, μπορούν να εμπλουτιστούν με πρόσθετες ικανότητες, όπως ύπαρξη «όρασης», τοπική βελτιστοποίηση, πράγματα που δεν συναντώνται στα πραγματικά μυρμήγκια.

Μια βασική διαφορά των τεχνητών μυρμηγκιών από τα πραγματικά μυρμήγκια είναι ότι εναποθέτουν φερομόνη όταν ολοκληρώσουν τα ταξίδια τους ενώ τα πραγματικά μυρμήγκια εναποθέτουν συνεχώς φερομόνη.

ΚΕΦΑΛΑΙΟ 4

ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ (ACO) ΓΙΑ ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ (TSP)

4.1 Εισαγωγή

Σύμφωνα με το πρόβλημα του περιοδεύοντος πωλητή (TSP), ένας πωλητής ξεκινά από μια πόλη (βάση) κατά τη διάρκεια της περιοδείας του και πρέπει να επισκεφθεί κάποιες πόλεις, που απέχουν μεταξύ τους κάποιες δεδομένες αποστάσεις, και στη συνέχεια να γυρίσει στην βάση του. Ζητείται να βρεθεί η σειρά με την οποία πρέπει να επισκεφθεί τις πόλεις αυτές ώστε το κόστος που θα προκύψει για τον πωλητή να είναι ελάχιστο. Κατά αναλογία μπορεί να ζητούνται οι απαιτούμενοι χρόνοι ή η μικρότερη δυνατή συνολική απόσταση.

Στη γενική του διατύπωση το πρόβλημα μοντελοποιείται με τη βοήθεια ενός πλήρους γραφήματος και εξετάζεται θεωρώντας ότι ο πωλητής πρέπει να επισκεφθεί μόνο μια φορά την κάθε πόλη. Πρόκειται για ένα Ευκλείδειο γράφημα, όπου ισχύει η ανισοϊσότητα του τριγώνου για οποιαδήποτε τριάδα πλευρών του τριγώνου. Στην περίπτωση αυτή το πρόβλημα ταυτίζεται με την εύρεση ενός Hamiltonian κύκλου με το ελάχιστο δυνατό βάρος.

Ο συνολικός αριθμός των Hamiltonian κύκλων σε ένα πλήρες γράφημα είναι $\frac{(n-1)!}{2}$. Αυτό προκύπτει εύκολα αν υποτεθεί ότι ο πωλητής ξεκινά από την αρχική πόλη και πρέπει να μεταβεί σε $n-1$ πόλεις, από εκεί σε $n-2$, σε $n-3, \mathbf{K}$, σε δύο (2) πόλεις και στη συνέχεια να επιστρέψει στη βάση του. Αυτές οι επιλογές είναι ανεξάρτητες και για αυτό προκύπτει ο όρος (n-1)! . Το πρόβλημα θα μπορούσε να λυθεί θεωρώντας όλους τους $\frac{(n-1)!}{2}$ Hamiltonian κύκλους και επιλέγοντας αυτόν με το ελάχιστο βάρος. Όμως, ακόμα και για μικρές τιμές του n το αποτέλεσμα θα οδηγούσε σε υπολογιστική έκρηξη.

Το πρόβλημα του περιοδεύοντος πωλητή χαρακτηρίζεται σαν «δύσκολο» (NP-hard) [Garey και άλλοι, 1972] και η αντιμετώπιση του με την κλασική μέθοδο του Δυναμικού Προγραμματισμού δεν δίνει αποτελεσματική λύση που να καταλήγει στην βέλτιστη λύση.

Οι μετα-ευρεστικοί αλγόριθμοι βελτιστοποίησης αποικίας μυρμηγκιών (ACO) εφαρμόστηκαν στο πρόβλημα του περιοδεύοντος πωλητή (TSP) και συγκρινόμενοι με άλλους ευρεστικούς αλγορίθμους έδωσαν πολύ ενδιαφέροντα και ικανοποιητικά αποτελέσματα.

4.2 Το πρόβλημα του περιοδεύοντος πωλητή (TSP)

Ο Dorigo και άλλοι (1996) επέλεξαν το πρόβλημα του περιοδεύοντος πωλητή (TSP) για να εφαρμόσουν μεθόδους βελτιστοποίησης, βασισμένες στις αποικίες μυρμηγκιών, για τους εξής λόγους:

i. είναι ένα πρόβλημα στο οποίο η αποικία μυρμηγκιών προσαρμόζεται εύκολα,

ii. είναι ένα NP-hard πρόβλημα συνδυαστικής βελτιστοποίησης το οποίο έχει μελετηθεί σε βάθος [Lawler και άλλοι, 1985], και

iii. ερμηνεύεται πολύ εύκολα έτσι ώστε η συμπεριφορά των αλγορίθμων αποικίας μυρμηγκιών να μην επιβαρύνεται με πολλές τεχνικές λεπτομέρειες.

Το πρόβλημα του περιοδεύοντος πωλητή εκφράζεται μέσω ενός πλήρους γραφήματος $G=(N,A)$ με N το σύνολο των κόμβων που αντιπροσωπεύουν τις πόλεις και A το σύνολο τον τόξων που συνδέουν πλήρως τους κόμβους του συνόλου N. Σε κάθε τόξο $(i,j)\in A$ αντιστοιχεί μια τιμή μήκους d_{ij}, η οποία είναι η απόσταση μεταξύ των πόλεων i και j με $i,j\in N$. Στην περίπτωση του ασύμμετρου προβλήματος του περιοδεύοντος πωλητή, η απόσταση μεταξύ δυο κόμβων i, j εξαρτάται από την διεύθυνση διάσχισης του τόξου και ισχύει $d_{ij}\neq d_{ji}$. Στο αντίστοιχο συμμετρικό TSP ισχύει $d_{ij}=d_{ji}$ για όλα τα τόξα που ανήκουν στο σύνολο A. Το TSP είναι το πρόβλημα εύρεσης του ελάχιστου μήκους του Hamiltonian κύκλου ενός γραφήματος, όπου Hamiltonian κύκλος είναι η κλειστή διαδρομή που διέρχεται μια και μόνο φορά από τους κόμβους $n=|N|$ του γραφήματος. Η διατύπωση ενός TSP είναι: δεδομένης μιας ακολουθίας n πόλεων

$c_1, c_2, \mathbf{K}, c_n$ [Colorni και άλλοι, 1992] ενώ σε κάθε ζευγάρι (c_i, c_j) αντιστοιχεί μια απόσταση $d(c_i, c_j)$, να βρεθεί μια μετάθεση π πόλεων έτσι ώστε να ελαχιστοποιηθεί η ποσότητα:

$$\sum_{i=1}^{n-1} d_{(c_{\pi(i)}, c_{\pi(i+1)})} + d_{(c_{\pi(n)}, c_{\pi(1)})} \qquad (4.1)$$

Ακολουθούν οι σημαντικότεροι αλγόριθμοι, βελτιστοποίησης, αποικίας μυρμηγκιών (ACO) που εφαρμόστηκαν με επιτυχία στο TSP.

4.3 Αλγόριθμοι βελτιστοποίησης αποικίας μυρμηγκιών (ACO) για το πρόβλημα του περιοδεύοντος πωλητή

Οι (ACO) αλγόριθμοι εφαρμόζονται απευθείας στο πρόβλημα του περιοδεύοντος πωλητή (TSP). Το γράφημα δομής $G = (C, L)$, όπου το σύνολο L συνδέει πλήρως τις συνιστώσες C, είναι πανομοιότυπο του γραφήματος του προβλήματος, δηλαδή, $C = N$ και $L = A$. Επομένως, C είναι οι συνιστώσες του γραφήματος και L τόξα που συνδέουν τις συνιστώσες. Οι περιορισμοί Ω ενισχύουν το γεγονός ότι τα μυρμήγκια βρίσκουν μόνο εφικτές διαδρομές που αντιστοιχούν σε μεταθέσεις των πόλεων. Αυτό είναι εφικτό, επειδή το γράφημα δομής είναι ένα πλήρες γράφημα και κάθε κλειστό μονοπάτι που επισκέπτεται όλους τους κόμβους, χωρίς να επαναλαμβάνει κάποιον από αυτούς, αντιστοιχεί σε μια εφικτή διαδρομή.

Στους αλγόριθμους (ACO) για το TSP, το μονοπάτι φερομόνης σχετίζεται με τα τόξα και για αυτό το λόγο η ποσότητα τ_{ij} αναφέρεται στην επιθυμία επίσκεψης της πόλης j αμέσως μετά την επίσκεψη της πόλης i. Η ευρεστική πληροφορία είναι $\eta_{ij} = 1/d_{ij}$, δηλαδή η ευρεστική επιθυμία του να πάει κάποιος από την πόλη i κατευθείαν στην πόλη j είναι αντιστρόφως ανάλογη της απόστασης μεταξύ των δυο πόλεων. Στην περίπτωση που είναι $d_{ij} = 0$, για κάποιο τόξο (i, j), η αντίστοιχη πληροφορία η_{ij} έχει πολύ μικρή τιμή.

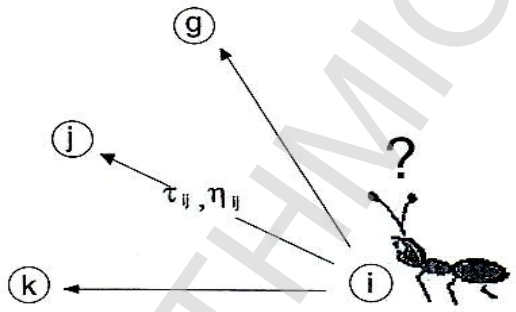
Οι διαδρομές σχηματίζονται από τα μυρμήγκια, εφαρμόζοντας την παρακάτω απλή διαδικασία:

i. Επιλογή, με κάποιο κριτήριο, μιας αρχικής πόλης στην οποία το μυρμήγκι τοποθετείται.

ii. Χρήση φερομόνης και ευρεστικών τιμών για το σχηματισμό της διαδρομής, σύμφωνα με κάποιο πιθανοκρατικό νόμο, προσθέτοντας σε κάθε επανάληψη τις πόλεις που το μυρμήγκι δεν έχει επισκεφτεί ακόμα, μέχρι να τις επισκεφτεί όλες (Σχήμα 4-1).

iii. Επιστροφή στην αρχική πόλη.

Αφού όλα τα μυρμήγκια έχουν ολοκληρώσει τις διαδρομές τους, στη συνέχεια εναποθέτουν φερομόνη στις διαδρομές που έχουν ακολουθήσει. Να σημειωθεί ότι σε μερικές περιπτώσεις, πριν από την εναπόθεση φερομόνης, οι διαδρομές που δημιουργούνται από τα μυρμήγκια μπορούν να βελτιωθούν εφαρμόζοντας κάποια τεχνική τοπικής αναζήτησης.



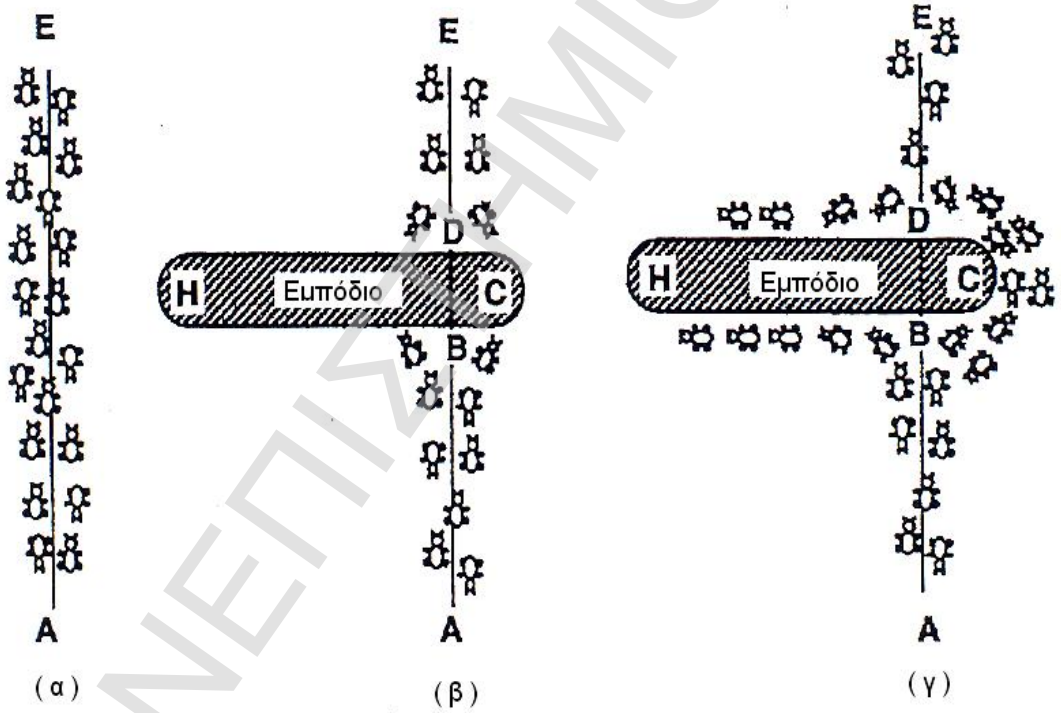
Σχήμα 4-1: Ένα μυρμήγκι που φτάνει στην πόλη i επιλέγει την επόμενη πόλη που θα μετακινηθεί σαν συνάρτηση των τιμών φερομόνης τ_{ij} και των ευρεστικών τιμών η_{ij} στα τόξα που συνδέουν την πόλη i με την πόλη j που το μυρμήγκι δεν έχει επισκεφθεί ακόμα.

Τα πραγματικά μυρμήγκια είναι ικανά να βρίσκουν:

i. Το συντομότερο μονοπάτι από μια πηγή τροφής προς τη φωλιά τους [Beckers, 1992], χωρίς να χρησιμοποιούν οπτικά μέσα [Halldobler και Wilson, 1990].

ii. Ένα νέο συντομότερο μονοπάτι όταν το παλιό δεν είναι πλέον εφικτό, εξαιτίας της τοποθέτησης ενός εμποδίου. Αυτό εξηγείται με βάση τη δυνατότητα που έχουν τα μυρμήγκια να προσαρμόζονται στις αλλαγές του περιβάλλοντος.

Θεωρείστε το Σχήμα 4-2(α): τα μυρμήγκια κινούνται σε μια ευθεία που συνδέει μια πηγή τροφής με τη φωλιά τους [Dorigo και Gambardella, 1996]. Είναι γνωστό ότι τα μυρμήγκια σχηματίζουν και διατηρούν την ευθεία μέσω του ίχνους φερομόνης. Κάθε μυρμήγκι αφήνει μια συγκεκριμένη ποσότητα φερομόνης καθώς περπατάει, ενώ με πιθανοκρατικό νόμο αποφασίζει να ακολουθήσει μια κατεύθυνση αυτούσια σε φερομόνη. Αυτή η συμπεριφορά των πραγματικών μυρμηγκιών χρησιμοποιείται για να εξηγήσει τον τρόπο που βρίσκουν το συντομότερο μονοπάτι στην περίπτωση που ένα αναπάντεχο εμπόδιο διακόψει την αρχική γραμμή (Σχήμα 4-2(β)). Στην πραγματικότητα, όταν εμφανιστεί το εμπόδιο, εκείνα τα μυρμήγκια που βρίσκονται ακριβώς μπροστά στο εμπόδιο δεν μπορούν να συνεχίσουν να ακολουθούν το ίχνος φερομόνης, οπότε πρέπει να επιλέξουν αν θα στρίψουν δεξιά ή αριστερά. Σε αυτή την περίπτωση, τα μισά μυρμήγκια επιλέγουν να στρίψουν δεξιά ενώ τα άλλα μισά αριστερά. Μια παρόμοια κατάσταση επικρατεί στην άλλη πλευρά του εμποδίου (Σχήμα 4-2(β)).



Σχήμα 4-2: Ένα παράδειγμα πραγματικών μυρμηγκιών. (α) Τα μυρμήγκια ακολουθούν μια διαδρομή μεταξύ των σημείων A και E. (β) Ένα εμπόδιο παρεμβάλλεται. Τα μυρμήγκια μπορούν να πάνε γύρω του, επιλέγοντας ένα από τα δύο διαφορετικά μονοπάτια με ίση πιθανότητα. (γ) Στο συντομότερο μονοπάτι εναποτίθεται περισσότερη ποσότητα φερομόνης.

Σημειώνετε ότι τα μυρμήγκια που διαλέγουν, τυχαία, το συντομότερο μονοπάτι γύρω από το εμπόδιο, θα ανασυγκροτήσουν πιο γρήγορα το ίχνος φερομόνης που είχε διακοπεί, σε σχέση με αυτά που διαλέγουν το μακρύτερο μονοπάτι. Έτσι, το συντομότερο μονοπάτι φιλοξενεί μεγαλύτερη ποσότητα φερομόνης ανά μονάδα χρόνου οπότε ένας μεγαλύτερος αριθμός μυρμηγκιών θα το επιλέξει. Λόγω αυτής της θετικής ανάδρασης (αυτο-καταλυτική διαδικασία), όλα τα μυρμήγκια θα επιλέξουν, γρήγορα, το συντομότερο μονοπάτι (Σχήμα 4-2(γ)). Η ενδιαφέρουσα πλευρά αυτής της αυτο-καταλυτικής διαδικασίας είναι ότι: η εύρεση του συντομότερου μονοπατιού γύρω από το εμπόδιο, είναι ιδιότητα που προκύπτει από την αλληλεπίδραση ανάμεσα:

i. Στο σχήμα του εμποδίου, και

ii. Στην κατανεμημένη συμπεριφορά των μυρμηγκιών. Ενώ όλα τα μυρμήγκια κινούνται με την ίδια περίπου ταχύτητα και εναποθέτουν ένα ίχνος φερομόνης με τον ίδιο περίπου ρυθμό, απαιτείται περισσότερος χρόνος να κάνουν το γύρο του εμποδίου από την μεγαλύτερη πλευρά του, παρά από την πιο μικρή, με αποτέλεσμα το ίχνος φερομόνης να συντηρείται στην μικρή πλευρά.

Από τα παραπάνω προκύπτει ότι η προτίμηση των μυρμηγκιών για υψηλότερα επίπεδα φερομόνης προκαλεί την συσσώρευσή της πιο γρήγορα στο συντομότερο μονοπάτι.

4.3-1 Ant System (AS)

Ο αλγόριθμος Ant System (AS) είναι πρόγονος όλων των ερευνητικών προσπαθειών που αφορούν τους αλγόριθμους μυρμηγκιών, και εφαρμόστηκε για πρώτη φορά στο TSP [Dorigo, 1992]. Οι αλγόριθμοι τους οποίους οι ερευνητές εμπνεύστηκαν από τον AS εμφανίστηκαν σαν ευρεστικές μέθοδοι οι οποίες βοηθούν στην επίλυση συνδυαστικών προβλημάτων βελτιστοποίησης.

Ο AS είναι ένας αλγόριθμος στον οποίο ένα σύνολο τεχνητών μυρμηγκιών συνεργάζονται για την επίλυση ενός προβλήματος, ανταλλάσσοντας πληροφορίες μέσω της φερομόνης που εναποθέτουν στο γράφημα. Τα κύρια χαρακτηριστικά του AS είναι:

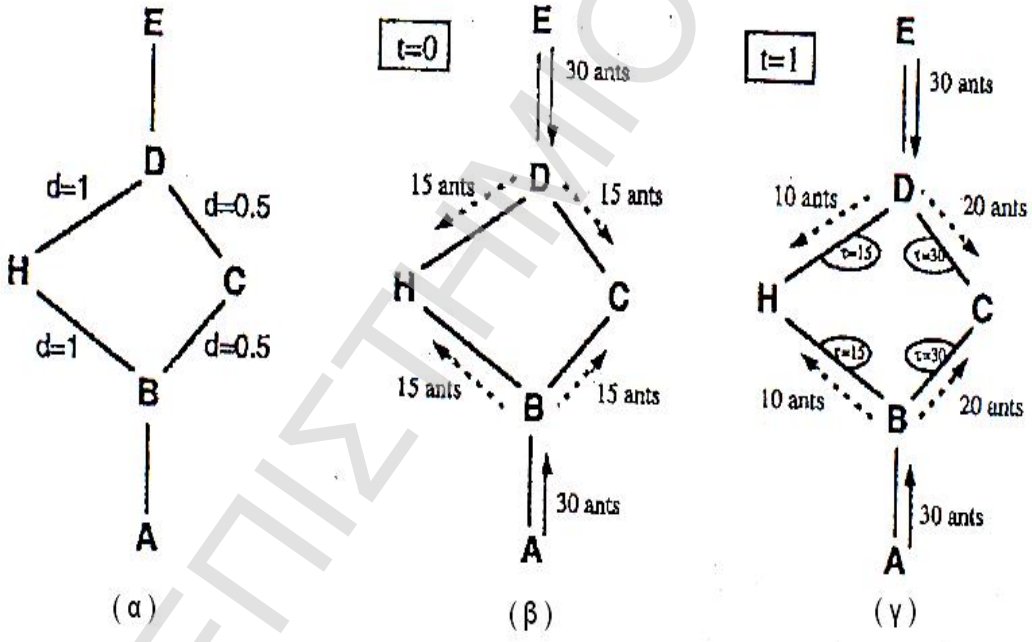
i. Ακολουθείται η θετική ανάδραση (επιτρέπει τη γρήγορη εύρεση των καλών λύσεων).

ii. Βασίζεται στον κατανεμημένο υπολογισμό (αποφεύγεται η πρόωρη σύγκλιση).

iii. Χρησιμοποιεί μια εποικοδομητική άπληστη ευρεστική συνάρτηση (που βοηθά στην εύρεση αποδεκτών λύσεων).

iv. Είναι μια βασισμένη στον πληθυσμό προσέγγιση [Aguilar, 1998].

Θεωρείστε το γράφημα του Σχήματος 4-3(α) το οποίο είναι μια ερμηνεία του AS της εικόνας του Σχήματος 4-2(β)



Σχήμα 4-3: Ένα παράδειγμα με τεχνητά μυρμήγκια. (α) Το αρχικό γράφημα με τις αποστάσεις. (β) Τη χρονική στιγμή $t=0$ δεν υπάρχει κανένα ίχνος φερομόνης στα τόξα του γραφήματος. Επομένως τα μυρμήγκια επιλέγουν να πάνε αριστερά ή δεξιά με ίση πιθανότητα. (γ) Τη χρονική στιγμή $t=1$ το ίχνος φερομόνης είναι πιο ισχυρό στα τόξα της συντομότερης διαδρομής και για αυτό το λόγο προτιμούνται από τα μυρμήγκια.

Υποθέστε ότι οι αποστάσεις μεταξύ D και H, μεταξύ B και H και μεταξύ B και D – μέσω C – είναι ίσες με 1 και έστω ότι το C είναι τοποθετημένο στο μέσο της διαδρομής από το B και D (Σχήμα 4-3(α)). Θεωρείστε τις διακριτές χρονικές επαναλήψεις $t = 0,1,2,\mathbf{K}$ Υποθέστε ότι 30 νέα μυρμήγκια φθάνουν στο B από το A και 30 στο D από το E σε κάθε χρονική στιγμή. Κάθε μυρμήγκι κινείται με ταχύτητα 1 ανά μονάδα χρόνου και καθώς προχωρά αφήνει στο έδαφος, ίχνος φερομόνης ίσο με 1 που, για χάρη απλότητας, εξατμίζεται τελείως.

Τη χρονική στιγμή $t = 0$ δεν υπάρχει κανένα ίχνος φερομόνης, αλλά 30 μυρμήγκια βρίσκονται στο B και 30 στο D. Η επιλογή του δρόμου που θα διαλέξουν πραγματοποιείται τυχαία. Επομένως, 15 μυρμήγκια από κάθε κόμβο θα περάσουν από τον H και 15 από τον C (Σχήμα 4-3(β)).

Τη στιγμή $t = 1$ τα νέα 30 μυρμήγκια που έρχονται στο B από το A βρίσκουν ένα ίχνος, έντασης 15, στη διαδρομή που οδηγεί στο H, το οποίο έχει τοποθετηθεί από τα 15 μυρμήγκια που περάσαν από εκεί, ξεκινώντας από το B, και ένα ίχνος, έντασης 30, στην διαδρομή προς το C, το οποίο προέκυψε από το άθροισμα του ίχνους φερομόνης που τοποθετήθηκε από τα 15 μυρμήγκια που περάσαν από εκεί, ξεκινώντας από το B, και από τα 15 μυρμήγκια που έφθασαν στο B, ερχόμενα από το D μέσω του C (Σχήμα 4-3(γ)). Επομένως, η πιθανότητα επιλογής μιας διαδρομής είναι αμερόληπτη, έτσι ώστε ο αναμενόμενος αριθμός μυρμηγκιών που θα περάσουν από C να είναι ο διπλάσιος από αυτών που θα περάσουν από το H: 20 και 10 αντίστοιχα. Το ίδιο ισχύει και για τα νέα 30 μυρμήγκια στο D, που έρχονται από το E.

Η διαδικασία αυτή συνεχίζεται μέχρι όλα τα μυρμήγκια να επιλέξουν τη συντομότερη διαδρομή.

Στα τεχνητά μυρμήγκια δίνονται μερικές δυνατότητες που δεν έχουν φυσικά πανομοιότυπα, αλλά οι οποίες έχει παρατηρηθεί ότι ταιριάζουν στην εφαρμογή του TSP: τα τεχνητά μυρμήγκια μπορούν να καθορίσουν πόσο μακριά θα είναι οι πόλεις. Είναι προικισμένα με λειτουργική μνήμη N^k (tabu list) που χρησιμοποιείται για την απομνημόνευση πόλεων τις οποίες έχουν ήδη επισκεφθεί. Η λειτουργική μνήμη αδειάζει στο ξεκίνημα κάθε νέας διαδρομής, και αναπροσαρμόζεται στην συνέχεια, για κάθε χρονικό βήμα, προσθέτοντας την νέα πόλη που επισκέφθηκαν.

Κάθε λύση του TSP δημιουργείται με τη διαδοχική μετάβαση των τεχνητών μυρμηγκιών από την μια πόλη στην άλλη με ένα πιθανοκρατικό κανόνα. Η διαδρομή ολοκληρώνεται όταν το τεχνητό μυρμήγκι επιστρέψει

στην βάση του (αρχική πόλη). Τότε βαθμολογείται η λύση που υλοποίησε με κριτήριο το συνολικό μήκος της διαδρομής οπότε προστίθεται η αντίστοιχη ποσότητα φερομόνης στο μονοπάτι που ακολούθησε. Αυτό γίνεται για κάθε μυρμήγκι της αποικίας μέχρι να συμπληρωθεί ο ζητούμενος αριθμός επαναλήψεων. Σημειώνεται, ότι μια επανάληψη (iteration) πραγματοποιείται όταν όλα τα μυρμήγκια ολοκληρώσουν τις διαδρομές τους.

Τα μυρμήγκια τοποθετούνται ή τυχαία στις πόλεις ή το καθένα σε μια διαφορετική πόλη στην αρχή της διαδρομής τους. Κάθε μυρμήγκι k κατά την επανάληψη t που βρίσκεται στην πόλη i επιλέγει την επόμενη πόλη j που θα επισκεφθεί με βάση ένα *τυχαίο – αναλογικό κανόνα μετάβασης* (random proportional transition rule), ο οποίος δίνεται:

$$P_{ij}^{k}=\begin{cases}\left[\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l\in N_i^k}\left[\tau_{il}(t)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}}\right] & , \quad \text{αν} \quad j\notin N_i^k \\ 0 & , \quad \text{αν} \quad j\in N_i^k\end{cases} \tag{4.2}$$

όπου:

N_i^k : είναι οι πόλεις που έχει ήδη επισκεφθεί το μυρμήγκι k, όταν βρίσκεται στην πόλη i (tabu list).

$\tau_{ij}(t)$: είναι η ποσότητα της φερομόνης που τοποθετείται στο τόξο που συνδέει τις πόλεις i και j και αντιπροσωπεύει την προτίμηση της επιλογής της πόλης j όταν το μυρμήγκι βρίσκεται στην πόλη i. Οι πληροφορίες από τα ίχνη φερομόνης αλλάζουν κατά την λύση του προβλήματος για να απεικονίζουν την εμπειρία που αποκτούν τα μυρμήγκια κατά την επίλυση του προβλήματος. Τα μυρμήγκια εναποθέτουν ποσότητα φερομόνης ανάλογα με την ποιότητα των λύσεων που παράγονται: όσο πιο σύντομες είναι οι διαδρομές που σχηματίζονται από ένα μυρμήγκι, τόσο μεγαλύτερο ποσό φερομόνης αυτό εναποθέτει στα τόξα που χρησιμοποίησε για να σχηματίσει τη διαδρομή.

n_{ij}: είναι η *ορατότητα* (visibility) και ορίζεται σαν το αντίστροφο της απόστασης μεταξύ των πόλεων i και j: $\eta_{ij} = \frac{1}{d_{ij}}$. Η ορατότητα εκφράζει την *ευρεστική προτίμηση* (heuristic desirability) της πόλης j σαν επόμενου σταθμού του μυρμηγκιού k όταν αυτό βρίσκεται στην πόλη i. Βασίζεται καθαρά σε τοπικές πληροφορίες.

Ο ρόλος των παραμέτρων α και β, που καθορίζονται από τον χρήστη και εξαρτώνται από το πρόβλημα προς επίλυση, είναι:

- Αν $\alpha = 0$, είναι πιθανότερο να επιλεγούν οι πιο κοντινές πόλεις, οπότε ο αλγόριθμος εκφυλίζεται σε ένα στοχαστικό αλγόριθμο με πολλά σημεία εκκίνησης γιατί τα μυρμήγκια αρχικά κατανέμονται τυχαία στους κόμβους.

- Αν $\beta = 0$, μόνο η ενίσχυση της φερομόνης είναι σε λειτουργία, οπότε η αναζήτηση της βέλτιστης λύσης στηρίζεται μόνο στη θετική ανάδραση. Αυτή η μέθοδος θα οδηγήσει στην γρήγορη εμφάνιση στασιμότητας, μιας κατάστασης όπου όλα τα μυρμήγκια κάνουν την ίδια διαδρομή, η οποία είναι κατά πολύ κατώτερη της άριστης [Dorigo και άλλοι, 1996].

Θα πρέπει να γίνεται κατάλληλη προσαρμογή των τιμών της ορατότητας και των τιμών ιχνών της φερομόνης αντίστοιχα.

Στον AS ο κανόνας της συνολικής εναπόθεσης (ανανέωσης) της φερομόνης την χρονική στιγμή t, εφαρμόζεται ως εξής: όταν το μυρμήγκι k ολοκληρώσει την διαδρομή του, κατά την επανάληψη $t+1$, η φερομόνη που εναποτίθεται ανάλογα σε όλα τα τόξα είναι (*κανόνας ανανέωσης της φερομόνης* – pheromone update rule):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4.3}$$

όπου ρ με $0 \le \rho < 1$ ο *συντελεστής εξάτμισης φερομόνης* με

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{4.4}$$

όπου $\Delta\tau_{ij}^{k}(t)$ είναι η ποσότητα της φερομόνης, ανά μονάδα μήκους, που εναποτίθεται στο τόξο (i,j) από το μυρμήγκι k μεταξύ των χρονικών στιγμών t και $t+1$, και δίνεται από τον τύπο:

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{L^{k}(t)}, & \text{αν} \quad (i,j)\in T^{k}(t)\\ 0 \quad , & \text{αν} \quad (i,j)\notin T^{k}(t)\end{cases} \tag{4.5}$$

όπου:

m: είναι ο συνολικός αριθμός μυρμηγκιών,

$T^{k}(t)$: είναι διαδρομή (ταξίδι) του μυρμηγκιού k κατά την επανάληψη t,

$L^{k}(t)$: είναι το συνολικό μήκος της διαδρομής $T^{k}(t)$,

Q: είναι μια παράμετρος που καθορίζεται από τον χρήστη η οποία δεν διαφοροποιεί την απόδοση του αλγορίθμου. Σε βελτιωμένες εκδόσεις του AS παραλείπεται.

Από την (4.5) προκύπτει ότι η ποσότητα της προστιθέμενης φερομόνης είναι αντιστρόφως ανάλογη της διανυόμενης απόστασης. Οπότε, συντομότερες διαδρομές είναι πιο ελκυστικές.

Έχουν διατυπωθεί αρκετές διαφορετικές εκδοχές για την βελτίωση του κλασικού αλγορίθμου AS [Gambardella και άλλοι, 1995]. Σε δύο από αυτές υπάρχουν αλγόριθμοι που εξαρτώνται από την πυκνότητα (ant-density) και την ποσότητα (ant-quantity) των μυρμηγκιών αντίστοιχα. Αυτές διαφέρουν ως προς τον τρόπο ενημέρωσης (ανανέωσης) της φερομόνης. Σε αυτά τα μοντέλα κάθε μυρμήγκι εναποθέτει το ίχνος του σε κάθε βήμα, χωρίς να περιμένει το τέλος της διαδρομής. Στο μοντέλο που εξαρτάται από την πυκνότητα των μυρμηγκιών μια ποσότητα Q φερομόνης εναποτίθεται στο τόξο (i,j) κάθε φορά που το μυρμήγκι πηγαίνει από τον κόμβο i στον κόμβο j. Στο μοντέλο που εξαρτάται από την ποσότητα των μυρμηγκιών, το μυρμήγκι εναποθέτει ποσότητα $Q/d_{(i,j)}$ ίχνους φερομόνης στο τόξο (i,j) κάθε φορά που κατευθύνεται από τον κόμβο i στον κόμβο j. Στο Σχήμα 4-4 παρουσιάζονται τα βήματα του κλασικού αλγορίθμου AS:

1 **For every** τόξο (i, j) **do** $\tau_{ij}(0) = c$;

2 **For** $k = 1$ **to** m **do:** τοποθέτησε το μυρμήγκι k σε μια τυχαία επιλεγμένη πόλη.

3 Έστω ότι T^t είναι η μικρότερη διαδρομή που έχει βρεθεί από την αρχή και L^t το μήκος αυτής

4 **For** $t = 1$ **to** t_{max} **do**

5 **For** $k = 1$ **to** m **do**

6 Σχημάτισε τη διαδρομή $T^k(t)$ εφαρμόζοντας $n-1$ φορές την **πιθανότητα μετάβασης**

$$P_{ij}^k(t) = \frac{\left[\tau_{ij}(t)\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum_{l \notin N_i^k} \left[\tau_{il}(t)\right]^\alpha \left[\eta_{il}\right]^\beta}$$

όπου i είναι η πόλη που βρίσκεται το μυρμήγκι

7 **End for**

8 **For** $k = 1$ **to** m **do**

9 Υπολόγισε το μήκος $L^k(t)$ της διαδρομής $T^k(t)$ που παράχθηκε από το μυρμήγκι k

10 **End for**

11 Αν μια βελτιωμένη διαδρομή βρεθεί τότε ενημέρωσε T^t και L^t

12 **For every** τόξο(i,j) **do**

13 Ενημέρωσε το μονοπάτι φερομόνης εφαρμόζοντας τον κανόνα:

$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$, όπου

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

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

14 **End for every**

15 **End for**

16 **End for**

17 **End for every**

18 **Print** T^t και L^t.

Σχήμα 4-4: Τα βήματα του αλγορίθμου AS.

Ο αλγόριθμος AS είναι ευέλικτος και ευπροσάρμοστος. Οπότε, αξιοποιείται άμεσα σε προβλήματα στα οποία η ορατότητα (ευρεστική πληροφορία) μεταβάλεται.

4.3-2 Ant Colony System (ACS)

Ο Ant Colony System είναι μια προέκταση του αλγορίθμου Ant System. Περιγράφεται στις εργασίες των Dorigo και Gambardella (1997). Τα συμπεράσματα τους υποδεικνύουν ότι αποτελεί μια αξιοσημείωτη βελτίωση του AS και ότι έχει καλύτερα αποτελέσματα για το πρόβλημα του περιοδεύοντος πωλητή (TSP) από ότι άλλοι αλγόριθμοι που ασχολούνται με το συγκεκριμένο πρόβλημα.

Ο κανόνας επιλογής της επόμενης πόλης που χρησιμοποιείται από τα μυρμήγκια έχει αλλάξει σε σχέση με τον AS. Κάποια χρονική στιγμή ένα μυρμήγκι θα επιλέξει μια πόλη χρησιμοποιώντας την ίδια τυχαία επιλογή όπως πριν. Τις υπόλοιπες φορές τα μυρμήγκια επιλέγουν μόνο την καλύτερη πόλη, βασισμένα στην ορατότητα και στο ίχνος φερομόνης. Η επιλογή μια από των δύο τεχνικών γίνεται με την βοήθεια ενός τυχαίου αριθμού, q, τον οποίο τον διαλέγουν τα μυρμήγκια πριν κάνουν την επιλογή της πόλης. Αν ο αριθμός αυτός είναι μικρότερος από μια σταθερά, q_0, τότε το μυρμήγκι θα επιλέξει την καλύτερη πόλη, αλλιώς θα επιλέξει μια τυχαία πόλη. Στους [Dorigo και Gambardella, 1997] αναφέρεται η επιλογή της καλύτερης πόλης σαν «εκμετάλλευση» (exploitation) και η επιλογή της αμερόληπτης τυχαίας πόλης σαν «αμερόληπτη εξερεύνηση» (biased exploration).

Ο αλγόριθμος ACS, γενικά, διαφέρει από τον AS ως προς τα εξής:

- *Ο κανόνας μετάβασης*, από κόμβο σε κόμβο, επιτρέπει την άμεση ύπαρξη ισορροπίας ανάμεσα στην εξερεύνηση των νέων τόξων και την εκμετάλλευση των προηγούμενων συσσωρευμένων πληροφοριών που αφορούν το πρόβλημα.

- *Ο κανόνας της συνολικής ανανέωσης της φερομόνης* εφαρμόζεται μόνο σε τόξα που ανήκουν στη διαδρομή που διάνυσε το καλύτερο μυρμήγκι.

- *Ο κανόνας της τοπικής ανανέωσης της φερομόνης* εφαρμόζεται καθώς τα μυρμήγκια «χτίζουν» τη λύση.

Κανόνας μετάβασης (transition rule)

Ένα τεχνητό μυρμήγκι k στην πόλη i επιλέγει την πόλη j, στην οποία θα μετακινηθεί ανάμεσα σε εκείνες τις πόλεις που δεν ανήκουν στη λειτουργική μνήμη N_i^k, εφαρμόζοντας τον παρακάτω πιθανοκρατικό κανόνα:

$$j = \begin{cases} \arg\max_{u \notin N_i^k} \left\{ \left[\tau_{iu}(t) \right] \left[\eta_{iu} \right]^{\beta} \right\} & , \quad \text{αν} \quad q \le q_0 \\ J & , \quad \text{αν} \quad q > q_0 \end{cases} \tag{4.6}$$

όπου:

$\tau_{iu}(t)$: είναι η ποσότητα της φερομόνης στο τόξο (i,u).

η_{iu}: είναι το αντίστροφο της απόστασης ανάμεσα στις πόλεις i και u.

J: είναι μια πιθανοκρατική μεταβλητή η οποία δίνει την πιθανότητα με την οποία ένα μυρμήγκι k στον κόμβο i επιλέγει τον κόμβο j, σύμφωνα με την παρακάτω εξίσωση:

$$P_{(i,j)}^k = \begin{cases} \dfrac{\left[\tau_{ij}(t) \right] \left[\eta_{ij} \right]^{\beta}}{\sum_{u \notin N_i^k} \left[\tau_{iu}(t) \right] \left[\eta_{iu} \right]^{\beta}}, & \text{αν} \quad j \notin N_i^k \\ 0 & , \quad \text{αλλιώς} \end{cases} \tag{4.7}$$

Κανόνας συνολικής ανανέωσης φερομόνης (global update rule)

Οι ποσότητες φερομόνης αλλάζουν τοπικά και συνολικά. Η συνολική ανανέωση φερομόνης αποσκοπεί στον εφοδιασμό με φερομόνη των συντομότερων διαδρομών.

Όταν τα τεχνητά μυρμήγκια ολοκληρώσουν τις διαδρομές τους, μόνο το μυρμήγκι που μέχρι αυτή τη χρονική στιγμή έχει κάνει την καλύτερη

διαδρομή εναποθέτει φερομόνη (καλύτερο μυρμήγκι), ενώ στον αλγόριθμο AS φερομόνη εναποθέτουν όλα τα μυρμήγκια. Η ποσότητα φερομόνης $\Delta\tau_{ij}(t)$ που εναποθέτει το καλύτερο μυρμήγκι σε κάθε τόξο (i,j) που επισκέφθηκε είναι αντιστρόφως ανάλογη του μήκους της διαδρομής: όσο πιο σύντομη είναι η διαδρομή τόσο μεγαλύτερη είναι η ποσότητα φερομόνης που εναπόθεσε το μυρμήγκι στα τόξα που συνθέτουν τη διαδρομή.

Ο κανόνας της συνολικής ανανέωσης φερομόνης δίνεται:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4.8}$$

όπου:

$$\Delta\tau_{ij}(t) = \begin{cases} \dfrac{1}{L_{best}(t)} & , \quad \text{αν} \quad (i,j) \in T_{best}(t) \\ 0 & , \quad \text{αν} \quad (i,j) \notin T_{best}(t) \end{cases} \tag{4.9}$$

όταν $L_{best}(t)$ είναι το μήκος της συνολικής βέλτιστης διαδρομής στη χρονική στιγμή t και $T_{best}(t)$ είναι η αντίστοιχη σειρά των πόλεων. Ο συντελεστής εξάτμισης φερομόνης, ρ, παίρνει τιμές στο διάστημα $[0,1)$. Η εξίσωση (4.8) υπαγορεύει ότι μόνο τα τόξα που ανήκουν στη συνολική βέλτιστη διαδρομή θα ενισχύονται με φερομόνη.

Κανόνας τοπικής ανανέωσης φερομόνης (local update rule)

Κατά τον σχηματισμό μιας διαδρομής, το μυρμήγκι αλλάζει την ποσότητα φερομόνης στα τόξα που έχει επισκεφθεί εφαρμόζοντας τον κανόνα της τοπικής ανανέωσης φερομόνης:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \tau_0 \tag{4.10}$$

όπου:

τ_0: είναι η αρχική ποσότητα φερομόνης.

Βρέθηκε, πειραματικά, [Dorigo και Gambardella, 1997], ότι $\tau_0 = (nL_{nn})^{-1}$ με n ο αριθμός πόλεων και L_{nn} το μήκος μιας διαδρομής που δημιουργείται με τον αλγόριθμο nearest neighborhood heuristic.

Το αποτέλεσμα της τοπικής ανανέωσης φερομόνης είναι να κάνει την προτίμηση των πόλεων να αλλάξει δυναμικά προκειμένου να αναπροσδιοριστεί η διαδρομή. Αν τα μυρμήγκια ερευνούν διαφορετικά μονοπάτια, τότε υπάρχει μεγαλύτερη πιθανότητα για ένα από αυτά να βρει μια βελτιωμένη λύση από το να ψάχναν όλα σε μια μικρού εύρους γειτονιά της προηγούμενης βέλτιστης διαδρομής. Κάθε φορά που ένα μυρμήγκι σχηματίζει ένα μονοπάτι, ο κανόνας τοπικής ανανέωσης μειώνει την ποσότητα φερομόνης των πόλεων που έχει επισκεφθεί και έτσι το μονοπάτι γίνεται λιγότερο ελκυστικό. Εξαιτίας αυτού, οι κόμβοι στη διαδρομή ενός μυρμηγκιού επιλέγονται με μικρότερη πιθανότητα, κατά τον σχηματισμό των διαδρομών των άλλων μηρμηγκιών. Κατά συνέπεια τα μυρμήγκια εκμεταλλεύονται τις πόλεις που δεν έχουν επισκεφθεί ακόμα και εμποδίζεται η σύγκλιση σε μία και μόνο μία διαδρομή.

Στον αλγόριθμο ACS γίνεται χρήση της *λίστας υποψήφιων πόλεων* (candidate list).

Λίστα Υποψήφιων Πόλεων (candidate list)

Μια *λίστα υποψήφιων πόλεων* είναι μια λίστα από πόλεις, μήκους cl (cl είναι αλγοριθμική παράμετρος), τις οποίες πρέπει το μυρμήγκι να επισκεφθεί, ξεκινώντας από μια δεδομένη αρχική πόλη. Οι πόλεις είναι ταξινομημένες σύμφωνα με το αντίστροφο της απόστασής τους και η προσπέλαση της λίστας γίνεται με σειριακό τρόπο. Αρχικά, ένα μυρμήγκι επιλέγει την επόμενη πόλη προς επίσκεψη από τις πόλεις που βρίσκονται στην λίστα χρησιμοποιώντας τον κλασικό κανόνα μετάβασης του αλγορίθμου ACS, για την επιλογή της επόμενης πόλης προς επίσκεψη. Από τη στιγμή που το μυρμήγκι έχει επισκεφτεί όλες τις κοντινές πόλεις, που ανήκουν στην λίστα, για μια δοσμένη πόλη i, επιλέγεται η επόμενη πόλη j από τις πιο κοντινές πόλεις που το μυρμήγκι δεν έχει επισκεφτεί ακόμα.

Στον Πίνακα 4-1 δείχνεται η σύγκριση του ACS με άλλους αλγορίθμους για το πρόβλημα του περιοδεύοντος πωλητή (TSP). Ο ACS εκτελέσθηκε για 1250 επαναλήψεις, χρησιμοποιώντας 20 μυρμήγκια.

Πρόβλημα	Ant Colony System (Dorigo & Gambardella, 1997)		Genetic Algorithms (Whitley και άλλοι,1989)		Simulated Annealing (Lin και άλλοι, 1993)	
	Μήκος Συντομότερης Διαδρομής	Αριθμός Διαδρομών	Μήκος Συντομότερης Διαδρομής	Αριθμός Διαδρομών	Μήκος Συντομότερης Διαδρομής	Αριθμός Διαδρομών
Eil 50 (50 πόλεις)	**425**	1830	428	25000	443	68512
Eil 75 (75 πόλεις)	**535**	3480	545	80000	580	173250
KroA 100 (100 πόλεις)	**21282**	4820	21761	103000	N/A	N/A

Πίνακας 4-1: Σύγκριση του αλγορίθμου ACS με άλλους αλγορίθμους για το πρόβλημα του περιοδεύοντος πωλητή (TSP).

4.3-3 Ant System – Elitist, AS_e

Η έννοια του ελιτισμού χρησιμοποιείται στους εξελικτικούς αλγορίθμους. Ο Dorigo και άλλοι (1996) αναπαρήγαγε και προσάρμοσε κατάλληλα αυτήν την ιδέα, στον αλγόριθμο Ant System (AS) με αποτέλεσμα να δημιουργηθεί ο αλγόριθμος των «εκλεκτών μυρμηγκιών» (elitist ants), AS_e. Η λειτουργία του κανόνα ενημέρωσης φερομόνης στον αλγόριθμο AS_e βασίζεται στην ιδέα ότι ένας επιπλέον αριθμός μυρμηγκιών, «εκλεκτών μυρμηγκιών» e, ενισχύει τη βέλτιστη διαδρομή T_{best} που βρέθηκε με την τρέχουσα επανάληψη t, προσθέτοντας ποσότητα φερομόνης ίσης με eQ/L_{best} . Η εξίσωση ενημέρωσης φερομόνης στον αλγόριθμο AS_e είναι:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) + e\Delta\tau^{e}_{ij}(t) \tag{4.11}$$

όπου:

$$\Delta\tau^{e}_{ij}(t) = \begin{cases} \dfrac{Q}{L_{best}(t)}, & \text{αν} \quad (i,j) \in T_{best}(t) \\ 0 \quad , & \text{αν} \quad (i,j) \notin T_{best}(t) \end{cases} \tag{4.12}$$

είναι $L_{best}(t)$ το μήκος της συνολικής βέλτιστης διαδρομής στο χρόνο t και $T_{best}(t)$ η αντίστοιχη σειρά των πόλεων.

4.3-4 Max-Min Ant System (MMAS)

Η έρευνα για τους αλγόριθμους ACO έδειξε ότι η βελτιωμένη απόδοση μπορεί να ληφθεί από μια ισχυρότερη εκμετάλλευση των καλύτερων λύσεων που βρίσκονται κατά τη διάρκεια της αναζήτησης. Χρησιμοποιώντας, όμως, μια πιο άπληστη αναζήτηση, ενδεχομένως, επιδεινώνεται το πρόβλημα της πρόωρης στασιμότητας της αναζήτησης. Οπότε, το κλειδί για να δημιουργηθεί καλύτερη απόδοση των αλγορίθμων ACO είναι ο συνδυασμός μιας βελτιωμένης εκμετάλλευσης των καλύτερων λύσεων που βρίσκονται κατά τη διάρκεια της αναζήτησης με έναν αποτελεσματικό μηχανισμό για την αποφυγή της πρόωρης στασιμότητας στην αναζήτηση. Ο αλγόριθμος Max-Min Ant System (MMAS) [Stützle και Hoos, 2000], αναπτύχθηκε για να καλύψει αυτές τις απαιτήσεις, διαφέρει σε τρία βασικά σημεία ως προς τον αντίστοιχο AS:

i. Για τη δημιουργία καλύτερων λύσεων, κατά τη διάρκεια μιας επανάληψης ή κατά την διάρκεια της εκτέλεσης του αλγορίθμου, μετά από κάθε επανάληψη μόνο ένα μεμονωμένο μυρμήγκι προσθέτει φερομόνη: αυτό που βρήκε την καλύτερη (βέλτιστη) λύση στην τρέχουσα επανάληψη (iteration best ant) ή αυτό που έχει βρει την συνολική βέλτιστη λύση (global best ant).

ii. Για την αποτροπή της στασιμότητας (stagnation) αναζήτησης οι τιμές της φερομόνης περιορίζονται σε ένα διάστημα $[\tau_{min}, \tau_{max}]$. Το άνω όριο, τ_{max}, δεν επιτρέπει την εμφάνιση στασιμότητας, κατά τη διάρκεια αναζήτησης, καθώς καμιά διαδρομή δεν μπορεί να συγκεντρώνει τόση φερομόνη ώστε να έλκει όλα τα μυρμήγκια. Το κάτω όριο, τ_{min}, εγγυάται ότι καμιά διαδρομή δεν θα έχει μηδενική ή περίπου μηδενική πιθανότητα επιλογής.

iii. Επιπλέον, αρχικοποιούνται, σκόπιμα, τα ίχνη φερομόνης σε τ_{max}, επιτυγχάνοντας μεγαλύτερη εξερεύνηση των λύσεων στην έναρξη του αλγορίθμου.

Στον αλγόριθμο MMAS ο τυχαιο-αναλογικός κανόνας μετάβασης (random – proportional transition rule), που καθορίζει τη μετάβαση του μυρμηγκιού κ, κατά την επανάληψη t, από την πόλη i στην πόλη j δίνεται:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l\in N_i^k}\left[\tau_{il}(t)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}}, & \text{αν} \quad j\notin N_i^k \\ 0 \qquad\qquad\qquad , & \text{αν} \quad j\in N_i^k\end{cases} \tag{4.13}$$

όπου:

N_i^k: είναι οι πόλεις που έχει ήδη επισκεφθεί το μυρμήγκι k όταν βρίσκεται στην πόλη i.

Στον αλγόριθμο MMAS μόνον ένα μυρμήγκι χρησιμοποιείται για την ενημέρωση της φερομόνης στο τέλος κάθε επανάληψης. Οπότε, ο κανόνας ανανέωσης φερομόνης δίνεται:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}(t) \tag{4.14}$$

όπου:

$$\Delta\tau_{ij}(t)=\Delta\tau_{ij}(t)=\begin{cases}\dfrac{1}{L_{(t)}^{*}} & , \quad \text{αν} \quad (i,j)\in T_{(t)}^{*} \\ 0 & , \quad \text{αν} \quad (i,j)\notin T_{(t)}^{*}\end{cases} \tag{4.15}$$

Όταν $L^{*}(t)$ είναι το μήκος της συνολικής βέλτιστης διαδρομής μέχρι τη χρονική στιγμή $t\left(L_{best}(t)\right)$ ή της καλύτερης λύσης κατά την επανάληψη t $\left(L_{iter}(t)\right)$ και $T^{*}(t)$ η σειρά των πόλεων αντίστοιχα. Το $L_{best}(t)$ και το $L_{iter}(t)$ μπορεί να εκφράζουν το κόστος της συνολικής βέλτιστης λύσης $\left(C_{best}(t)\right)$ και της καλύτερης λύσης $\left(C_{iter}(t)\right)$ σε κάθε επανάληψη. Η χρήση μιας μόνο λύσης, συνολικής βέλτιστης λύσης ή της καλύτερης λύσης σε κάθε επανάληψη, για την ανανέωση της φερομόνης είναι το πιο σημαντικό μέσο εκμετάλλευσης της αναζήτησης στον MMAS. Με μια συνετή επιλογή μεταξύ συνολικής βέλτιστης λύσης και καλύτερης λύσης σε κάθε επανάληψη, για την ανανέωση της φερομόνης ελέγχεται ο τρόπος που η εμπειρία της αναζήτησης αξιοποιείται. Όταν χρησιμοποιείται η συνολική βέλτιστη λύση, η αναζήτηση μπορεί να συγκεντρωθεί πάρα πολύ γρήγορα γύρω από αυτήν την λύση και η εξερεύνηση, πιθανώς, καλυτέρων λύσεων είναι περιορισμένη, με κίνδυνο ο αλγόριθμος να παγιδευτεί σε κακής ποιότητας λύσεις.

Αυτός ο κίνδυνος μειώνεται όταν επιλέγεται η καλύτερη λύση σε κάθε επανάληψη για την ανανέωση της φερομόνης όταν αυτή διαφέρει σημαντικά από επανάληψη σε επανάληψη και ένας μεγαλύτερος αριθμός συστατικών της λύσης μπορούν να πάρουν περιστασιακή ενίσχυση. Μπορούν να χρησιμοποιηθούν μικτές τεχνικές: όπως, επιλογή της καλύτερης λύσης σε κάθε επανάληψη ως προεπιλογή για την ανανέωση της φερομόνης και τη χρησιμοποίηση της συνολικής βέλτιστης λύσης μόνο σε κάθε σταθερό αριθμό επαναλήψεων. Όταν χρησιμοποιείται ο αλγόριθμος MMAS με τοπική αναζήτηση στην επίλυση του TSP, η καλύτερη τεχνική φαίνεται να είναι η χρήση μιας δυναμικής, μικτής τεχνικής που αυξάνει την συχνότητα της χρησιμοποίησης της συνολικής βέλτιστης λύσης για την ανανέωση της φερομόνης και την διάρκεια της αναζήτησης.

Ανεξαρτήτως της επιλογής μεταξύ της συνολικής βέλτιστης λύσης και της καλύτερης λύσης σε κάθε επανάληψη για την ανανέωση της φερομόνης, μπορεί να προκύψει στασιμότητα της αναζήτησης. Αυτό μπορεί να συμβεί αν σε κάθε ένα σημείο επιλογής, η ποσότητα της φερομόνης είναι σημαντικά μεγαλύτερη για μια επιλογή από ό,τι για όλες τις άλλες. Στην περίπτωση του TSP, αυτό σημαίνει ότι για κάθε πόλη, ένα από τα τόξα που εξέρχονται έχει ένα πολύ πιο υψηλό επίπεδο φερομόνης από τα άλλα. Σε αυτήν την κατάσταση, λόγω του πιθανοκρατικού τρόπου επιλογής ένα μυρμήγκι θα προτιμήσει αυτό το τμήμα της λύσης, από όλες τις εναλλακτικές λύσεις με αποτέλεσμα να δοθεί περαιτέρω ενίσχυση στην ανανέωση της φερομόνης. Σε μια τέτοια κατάσταση τα μυρμήγκια κατασκευάζουν συνεχώς την ίδια λύση και η εξερεύνηση σταματά.

Η στασιμότητα πρέπει να αποτραπεί. Ένας τρόπος για να γίνει αυτό, είναι να επηρεαστούν οι πιθανότητες για την επιλογή του επόμενου τμήματος λύσης, που εξαρτώνται άμεσα από την ποσότητα φερομόνης και τις ευρεστικές πληροφορίες. Οι ευρεστικές πληροφορίες εξαρτώνται από την φύση του προβλήματος και είναι στατικές σε όλο τον αλγόριθμο. Αλλά με τον περιορισμό των ποσοτήτων της φερομόνης, η στασιμότητα, μπορεί να αποτραπεί: οι σχετικές διαφορές μεταξύ των ποσοτήτων της φερομόνης να γίνουν πολύ ακραίες κατά την διάρκεια εκτέλεσης του αλγορίθμου. Για την υλοποίηση αυτού του στόχου, ο MMAS επιβάλλει σαφή όρια τ_{min} και τ_{max} στις ελάχιστες και στις μέγιστες ποσότητες φερομόνης έτσι ώστε για την φερομόνη $\tau_{ij}(t)$ να ισχύει: $\tau_{min} \le \tau_{ij}(t) \le \tau_{max}$. Στο τέλος κάθε επανάληψης πρέπει να εξασφαλιστεί ότι η φερομόνη βρίσκεται μέσα στα όρια. Αν είναι $\tau_{ij}(t) > \tau_{max}$, τότε $\tau_{ij}(t) = \tau_{max}$. Αν είναι $\tau_{ij}(t) < \tau_{min}$, τότε $\tau_{ij}(t) = \tau_{min}$. Αν είναι $\tau_{min} > 0$ και $\eta_{ij} < \infty$ για όλα τα τμήματα λύσης, η

πιθανότητα επιλογής ενός συγκεκριμένου τμήματος λύσης δεν είναι ποτέ μηδέν.

Πρέπει να επιλεγούν οι κατάλληλες τιμές των ορίων της φερομόνης.

Εύρεση της τιμής τ_{max}:

Σε κάθε επανάληψη η μέγιστη ποσότητα φερομόνης που μπορεί να εναποτεθεί ισούται με $\frac{1}{C_{opt}}$, όπου C_{opt} είναι η βέλτιστη λύση για ένα δεδομένο πρόβλημα.

Από την εξίσωση (4.14) ανανέωσης φερομόνης, για $t = 0,1,2,\mathbf{K},t$, προκύπτει:

για $t = 0$, $\quad \tau_{ij}(1) = (1-\rho)\tau_{ij}(0) + \frac{1}{C_{opt}}$

για $t = 1$ $\quad \tau_{ij}(2) = (1-\rho)^2 \tau_{ij}(0) + \frac{1-\rho}{C_{opt}} + \frac{1}{C_{opt}}$

για $t = 2$, $\quad \tau_{ij}(3) = (1-\rho)^3 \tau_{ij}(0) + \frac{(1-\rho)^2}{C_{opt}} + \frac{1-\rho}{C_{opt}} + \frac{1}{C_{opt}}$

M

για $t = t$, $\quad \tau_{ij}(t) = (1-\rho)^t \tau_{ij}(0) + \frac{1}{C_{opt}} \sum_{i=0}^{t-1} (1-\rho)^i$ (4.16)

είναι:

$$\lim_{t\to\infty} \tau_{ij}(t) = \lim_{t\to\infty} (1-\rho)^t \tau_{ij}(0) + \frac{1}{C_{opt}} \lim_{t\to\infty} \sum_{i=0}^{t-1} (1-\rho)^i ,$$

ισχύει $\rho < 1$, οπότε:

$$\lim_{t\to\infty} (1-\rho)^t = 0 ,$$

και

$$\lim_{t\to\infty} \sum_{i=0}^{t-1} (1-\rho)^i = \frac{1}{\rho}$$

Τελικά, είναι:

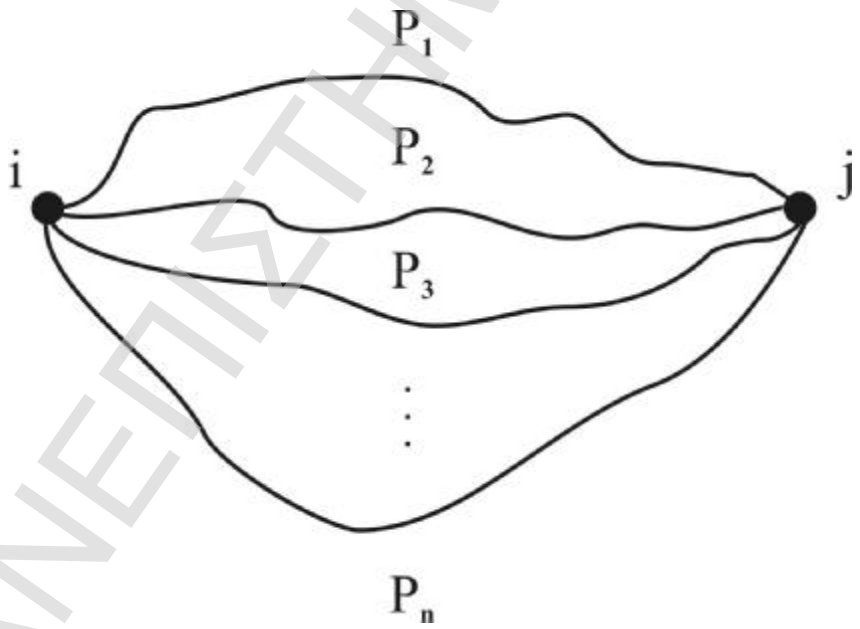
$$\lim_{t\to\infty} \tau_{ij}(t) = \frac{1}{\rho C_{opt}}$$

και για κάθε τ_{ij} ισχύει:

$$\tau_{ij}(t) \le \frac{1}{\rho C_{opt}} = \tau_{max}$$

Εύρεση της τιμής τ_{min}:

Στο TSP η βέλτιστη λύση του προβλήματος δημιουργείται, αν σε κάθε πόλη το μυρμήγκι επιλέξει την διαδρομή με την μέγιστη φερομόνη, τ_{max} Έστω ότι η πιθανότητα επιλογής της σωστής διαδρομής, σε όλες τις πόλεις είναι ίδια και ίση με P_{dec} (Σχήμα 4-5).



Σχήμα 4-5: Η πιθανότητα επιλογής της σωστής διαδρομής σε όλες τις πόλεις είναι ίδια και ισούται με P_{dec} .

Αν η πιθανότητα βέλτιστης λύσης είναι P_{best}, τότε:

$$P_{best} = P_1 P_2 P_3 \mathbf{K} P_n,$$

αλλά $P_1 = P_2 = P_3 = \mathbf{L} = P_n = P_{dec}$,

οπότε $P_{best} = P_{dec}^n$,

και $P_{dec} = \sqrt[n]{P_{best}}$ (4.17)

Για ορατότητα ίση με την μονάδα και για πλήθος τόξων $n/2$ που συνδέουν την πόλη που βρίσκεται, το μυρμήγκι, με τις υπόλοιπες πόλεις, ο πιθανοκρατικός κανόνας μετάβασης γίνεται:

$$\sqrt[n]{P_{best}} = \frac{\tau_{max}}{\tau_{max} + \left(\frac{n}{2} - 1\right)\tau_{min}} \quad (4.18)$$

Είναι:

$$\sqrt[n]{P_{best}}\,\tau_{max} + \sqrt[n]{P_{best}}\left(\frac{n}{2} - 1\right)\tau_{min} = \tau_{max}$$

και

$$\tau_{min} = \frac{\tau_{max}\left(1 - \sqrt[n]{P_{best}}\right)}{\left(\frac{n}{2} - 1\right)\sqrt[n]{P_{best}}} \quad (4.19)$$

Είναι:

$$\frac{\tau_{min}}{\tau_{max}} = \frac{1 - \sqrt[n]{P_{best}}}{\left(\frac{n}{2} - 1\right)\sqrt[n]{P_{best}}} \le 1,$$

οπότε

$$n \geq \frac{2}{\sqrt[n]{P_{best}}},$$

και

$$1 \leq \frac{n}{2}\sqrt[n]{P_{best}} \quad , \quad 2 \leq n\sqrt{P_{best}},$$

οπότε

$$n \geq \frac{2}{\sqrt[n]{P_{best}}} \tag{4.20}$$

Από την εξίσωση (4.19) προκύπτει ότι για να είναι $\tau_{min} \leq \tau_{max}$ πρέπει να ισχύει η σχέση (4.20).

Αρχικοποιώντας τη φερομόνη στην τιμή τ_{max} μέσω της ρύθμισης $\tau_0 = \tau_{max}$ ενισχύεται η εξερεύνηση (exploration) του χώρου των λύσεων στις πρώτες επαναλήψεις του αλγορίθμου, με αποτέλεσμα να βελτιώνεται η απόδοση του MMAS. Όταν προστεθεί φερομόνη σε ομοιόμορφο χώρο με πολύ χαμηλά επίπεδα φερομόνης τότε η διαφορά των επιπέδων της φερομόνης μεταξύ των καλών και των κακών λύσεων είναι μεγάλη. Τότε ο αλγόριθμος οδηγείται στην εκμετάλλευση (exploitation) του χώρου των καλών λύσεων σε βάρος της εξερεύνησης όλου του χώρου των λύσεων.

Ένας πρόσθετος μηχανισμός, αποκαλούμενος «Ομαλοποίηση της φερομόνης» (pheromone trail smoothing) μπορεί να χρησιμοποιηθεί για την αύξηση της απόδοσης του MMAS.

Ο μηχανισμός αυτός αυξάνει τα επίπεδα φερόμονης αναλογικά με την διαφορά τους από το τ_{max}, ώστε να αυξηθεί η πιθανότητα επιλογής των διαδρομών με χαμηλά επίπεδα φερομόνης.

Η μαθηματική υλοποίηση του παραπάνω μηχανισμού είναι:

$$\tau_{ij}^{*}(t) = \tau_{ij}(t) + \delta\left(\tau_{max}(t) - \tau_{ij}(t)\right) \tag{4.21}$$

όπου $\tau_{ij}(t)$ και $\tau_{ij}^{*}(t)$ είναι η ποσότητα φερομόνης πριν και μετά από την υλοποίηση. Η παράμετρος δ καθορίζεται από το χρήστη. Είναι $0 \leq \delta \leq 1$. Όταν $\delta < 1$ η πληροφορία που συλλέγεται κατά τη διάρκεια της εκτέλεσης

του αλγορίθμου δεν χάνεται εντελώς αλλά μόνον αποδυναμώνεται. Για $\delta = 1$, αυτός ο μηχανισμός αντιστοιχεί σε μια επαναρχικοποίηση της φερομόνης, ενώ για $\delta = 0$ απενεργοποιείται.

Ο παραπάνω μηχανισμός έχει μεγάλη αξία όταν ο αλγόριθμος εκτελείται με πολύ μεγάλο αριθμό επαναλήψεων με αποτέλεσμα να εξερευνάται καλύτερα ο χώρος των λύσεων. Συγχρόνως, κάνει τον MMAS λιγότερο ευαίσθητο στην ιδιαίτερη επιλογή του χαμηλότερου ορίου φερομόνης, τ_{min}.

4.3-5 Αλγόριθμος AS_{rank}

Ο Bullnheiner και άλλοι (1999) πρότειναν μια τροποποίηση του AS. Ο αλγόριθμος που προέκυψε ονομάστηκε AS_{rank}. Στον αλγόριθμο AS, η ανανέωση της φερομόνης γίνεται στο τέλος κάθε διαδρομής. Η κύρια διαφορά είναι ότι στον προτεινόμενο αλγόριθμο οι ευρεθείσες λύσεις ταξινόμουνται σε μια φθίνουσα σειρά ανάλογα με την ποιότητά τους στο τέλος κάθε διαδρομής και στη συνέχεια μόνο τα τόξα που ανήκουν στις πρώτες $\omega - 1$ λύσεις θα λάβουν μια ποσότητα φερομόνης, ανάλογη της κατάταξης της λύσης. Επιπλέον, μια επιπρόσθετη ποσότητα φερομόνης (ισοδύναμη με τη φερομόνη των «εκλεκτών μυρμηγκιών») θα προστεθεί στα τόξα που ανήκουν στη βέλτιστη διαδρομή. Η ανανέωση της φερομόνης [Panta Lucic, 2002] δίνεται από την παρακάτω εξίσωση:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \omega\Delta\tau_{ij}(t,t+1) + \Delta\tau_{ij}^{r}(t,t+1) \qquad (4.22)$$

όπου:

$\Delta\tau_{ij}(t,t+1)$: είναι η επιπρόσθετη φερομόνη στην βέλτιστη διαδρομή

$\Delta\tau_{ij}^{r}(t,t+1)$: είναι η επιπρόσθετη φερομόνη που θα εναποτεθεί σε κάθε τόξο που ανήκει σε μία ή περισσότερες διαδρομές στις πρώτες $\omega - 1$ διαδρομές (ταξινομημένες κατά μήκος) με

$$\Delta\tau_{ij}^{r}(t,t+1)=\sum_{\xi=1}^{\omega-1}\Delta\tau_{ij}^{\xi}(t) \qquad (4.23)$$

και $$\Delta\tau_{ij}^{\xi}(t)=\begin{cases}\dfrac{(\omega-\xi)}{L^{\xi}(t)} & , \text{ αν το μυρμήγκι με τάξη } \xi \text{ έχει}\\ & \text{χρησιμοποιήσει το τόξο } (i,j) \text{ στη}\\ & \text{διαδρομή του}\\ 0 & , \text{ αλλιώς}\end{cases}$$

όπου:

$L^{\xi}(t)$: είναι το μήκος της διαδρομής με τάξη ξ.

Στον Πίνακα 4-2 παρουσιάζονται τα συγκριτικά αποτελέσματα για το TSP των γνωστών αλγορίθμων: AS, ACS, AS_e, AS_{rank}, MMAS, [Stützle και Hoos, 2000]

Πρόβλημα	**Βέλτιστη**	**MMAS+pts**	**MMAS**	**ACS**	**AS_{rank}**	**AS_{rank}+pts**	**ASe**	**ASe+pts**	**AS**
eil51	426	**427.1**	427.6	428.1	434.5	428.8	428.3	427.4	437.3
kroA100	21282	***21291.6***	21320.3	21420.0	21746.0	21394.9	21522.8	21431.9	22471.4
d198	15780	**15956.8**	15972.5	16054.0	16199.1	16025.2	16205.0	16140.8	16702.1
ry48p	14422	14523.4	14553.2	14565.4	**14511.4**	14644.6	14685.2	14657.9	15296.4
ft70	38673	**38922.7**	39040.2	39090.0	39410.1	39199.2	39261.8	39161.0	39596.3
kro124p	36230	**36573.6**	36773.5	36857.0	36973.5	37218.0	37510.2	37417.7	38733.1
ftv170	2755	**2817.7**	2828.8	2826.5	2854.2	2915.6	2952.4	2908.1	3154.5

Πίνακας 4-2: Σύγκριση των αλγορίθμων AS, ACS, AS_e, AS_{rank}, MMAS στο TSP. Στον πίνακα εμφανίζονται οι μέσοι όροι ενός αριθμού εκτελέσεων κάθε αλγορίθμου. Η στήλη «βέλτιστη» εκφράζει τις βέλτιστες λύσεις κάθε προγράμματος.

Τα προβλήματα eil51, kroA100, d198 αναφέρονται στο συμμετρικό TSP και τα προβλήματα ry48p, ft70, kro124p, ftv170 στο ασύμμετρο TSP, όπως προτάθηκαν απο τον Stützle στον πρώτο διεθνή διαγωνισμό για την

εξελικτική βελτιστοποίηση (First International Countest on Evolutionary Optimization). Όλοι οι αλγόριθμοι χρησιμοποιούν τον ίδιο αριθμό διαδρομών για τα μυρμήγκια. Ο αριθμός αυτός ορίζεται ίσος με $k \cdot n \cdot 1000$, όπου $\kappa = 1$ για το συμμετρικό TSP και $\kappa = 2$ για το ασύμμετρο TSP και n ο αριθμός των πόλεων του προβλήματος.

Για τον αλγόριθμο MMAS χρησιμοποιήθηκαν οι εξής παράμετροι: $\alpha = 1,\ \beta = 2,\ \rho = 0,02,\ m = n,\ f_{best} = 0,05,\ \delta = 0,5$. Η ανανέωση της φερομόνης γίνεται μόνον απο το μυρμήγκι με την καλύτερη λύση σε κάθε επανάληψη, ενώ χρησιμοποιήθηκε λίστα υποψήφιων πόλεων. Για τον αλγόριθμο AS οι παράμετροι που χρησιμοποίηθηκαν είναι: $\alpha = 1,\ \beta = 5,\ \rho = 0,5,\ m = n\ Q = 100,\ \tau_0 = 10^{-6}$. Για τον AS_e είναι: $\alpha = 1,\ \beta = 5,\ \rho = 0,5,\ m = e = n,\ Q = 100,\ \tau_0 = 10^{-6}$. Χρησιμοποιήθηκε ο μηχανισμός ομαλοποίησης φερομόνης (tpts) για το συμμετρικό TSP με $\beta = 1$ και $\beta = 2$ για το ασύμμετρο TSP, ενώ η παράμετρος δ ισούται με 0,5. Οι αντίστοιχοι παράμετροι για τους αλγορίθμους ACS και AS_{rank} είναι: για τον αλγόριθμο ACS: $\alpha = 1,\ \beta = 2,\ \rho = 0,1,\ q_0 = 0,9,\ cl = 15,\ m = 10$ και για τον αλγόριθμο AS_{rank}: $\alpha = 1,\ \beta = 2,\ e = 6$. Χρησιμοποιώντας το μηχανισμό ομαλοποίησης φερομόνης (tpts) για το συμμετρικό TSP τότε είναι $\beta = 1$, ενώ για το ασύμμετρο TSP $\beta = 2$ ενώ $\delta = 0,5$.

4.4 Παράλληλες εφαρμογές

Οι αλγόριθμοι ACO εμπνέονται από τα πληθυσμιακά χαρακτηριστικά μιας αποικίας μυρμηγκιών οπότε είναι ιδιαίτερα κατάλληλοι για παράλληλες εφαρμογές. Υπάρχουν, θεωρητικά, τρεις διαφορετικοί τύποι παραλληλισμού [Corne και άλλοι, 1999]:

i. Παραλληλισμός στο επίπεδο των μυρμηγκιών.

ii. Παραλληλισμός στο επίπεδο των δεδομένων, και

iii. Συναρτησιακός παραλληλισμός.

Ο παραλληλισμός στο επίπεδο των μυρμηγκιών, που είναι ο πιο προφανής τρόπος παραλληλισμού ενός πληθυσμού ACO, εξετάζει ένα αριθμό αποικιών, κάθε μία από τις οποίες εφαρμόζεται στην ίδια περίπτωση του προβλήματος. Μπορεί να επιτρέπεται ή και να μην επιτρέπεται στις

αποικίες μυρμηγκιών να ανταλλάσουν πληροφορίες για τη διαδικασία αναζήτησης (λύση του προβλήματος).

Ο παραλληλισμός στο επίπεδο των δεδομένων, περιλαμβάνει την διάσπαση του εξεταζόμενου προβλήματος σε ένα αριθμό υποπροβλημάτων στον τομέα των δεδομένων, καθένα από τα οποία επιλύεται από μια αποικία μυρμηγκιών.

Ο συναρτησιακός παραλληλισμός, μπορεί εύκολα να υλοποιηθεί, σε έναν αλγόριθμο ACO, απλώς αφήνοντας τις διαδικασίες: δημιουργία και δραστηριότητα μυρμηγκιών, εξάτμιση φερομόνης και ενέργειες daemon να εκτελούν τις δραστηριότητές τους ταυτόχρονα, ίσως, ανταλλάσσοντας σήματα συγχρονισμού. Ο συναρτησιακός παραλληλισμός μπορεί να συνδυαστεί με τους δύο άλους τύπους παραλληλισμού. Προς το παρόν, όλες οι παράλληλες εφαρμογές των αλγορίθμων ACO χρησιμοποιούν τον παραλληλισμό στο επίπεδο των μυρμηγκιών [Bolandi & Bondanza, 1993].

Πρόσφατα, από τους Bullnheimer και άλλους (1997) προτάθηκαν η Συγχρονισμένη Παράλληλη Εφαρμογή (Synchronous Parallel Implementation) (SPI) για τον αλγόριθμο AS και η Μερικώς Ασύγχρονη Παράλληλη Εφαρμογή (Partially Asynchronous Parallel Implementation) (PAPI). Οι δυο αλγόριθμοι έχουν αξιολογηθεί με προσομοίωση. Τα ποτελέσματα έδειξαν ότι ο PAPI έχει καλύτερη απόδοση απο τον SPI, όταν η απόδοση μετριέται σύμφωνα με το χρόνο ροής και την επιτάχυνση. Αυτό οφείλεται πιθανώς, στην μειωμένη επικοινωνία που υπάρχει στον PAPI που είναι αποτέλεσμα της λιγότερο συχνής ανταλλαγής πληροφοριών από τα ίχνη φερομόνης ανάμεσα στις υπο-αποικίες. Απαιτούνται περισσότερα πειράματα για να συγκριθεί η ποιότητα των αποτελεσμάτων που παράγονται από τις εφερμογές SPI και PAPI.

4.5 Το πλήθος των μυρμηγκιών μιας αποικίας

Ο ακριβής αριθμός των μυρμηγκιών που χρησιμοποιείται για την επίλυση ενός προβλήματος είναι μια χαρακτηριστική παράμετρος που πρέπει να οριστεί πειραματικά. Προκύπτει, όμως, το εξής ερώτημα: Γιατί απαιτείται η χρήση μιας αποικίας μυρμηγκιών αντί ενός μυρμηγκιού; Στην πραγματικότητα, αν και ένα μυρμήγκι είναι σε θέση να παράγει μια λύση, οι εκτιμήσεις αποδοτικότητας προτείνουν ότι η χρήση της αποικίας των μυρμηγκιών είναι, συχνά, μια επιθυμητή επιλογή. Αυτό είναι ιδιαίτερα αληθές για προβλήματα συνδυαστικής βελτιστοποίησης, όπως το πρόβλημα

δρομολόγησης οχημάτων, στο οποίο τα μυρμήγκια λύνουν πολλά προβλήματα διαδρομών παράλληλα, οπότε πρέπει να χρησιμοποιηθεί μια αποικία για καθένα από αυτά τα προβλήματα.

Εμπειρικά στοιχεία δείχνουν ότι η απόδοση ενός αλγορίθμου βρίκεται στο βέλτιστο σημείο της όταν ένας αριθμός m μυρμηγκιών ορίζεται στην τιμή $M > 1$, όπου η παράμετρος M εξαρτάται, γενικά, από την κατηγορία των προβλημάτων στην οποία εφαρμόζεται ο αλγόριθμος.

4.6 Σύγκριση των αλγορίθμων ACO

Τα αποτελέσματα ερευνών στους αλγορίθμους ACO, όσον αφορά την απόδοσή τους, δείχνουν ότι ο αλγόριθμος MMAS επιτυγχάνει, έντονα, βελτιωμένη απόδοση έναντι του AS και άλλων βελτιωμένων εκδόσεων του AS. Ο αλγόριθμος MMAS βασίζεται σε διάφορες τροποποιήσεις του AS που στοχεύουν:

i. Να εκμεταλλευτούν εντονότερα τις καλύτερες λύσεις που βρίσκονται κατά τη διάρκεια αναζήτησης και για να κατευθύνουν την αναζήτηση των μυρμηγκιών στις λύσεις υψηλής ποιότητας, και

ii. Να αποτραπεί η πρόωρη σύγκλιση της αναζήτησης των μυρμηγκιών.

Μια από τις κύριες ιδέες που εισάγονται από τον αλγόριθμο MMAS είναι η χρησιμοποίηση των ορίων ιχνών φερομόνης για να αποτραπεί η πρόωρη σύγκλιση. Τα όρια αυτά μπορούν να εφαρμοστούν με διαφορετικό τρόπο, που μπορεί να ερμηνευθεί σαν υβριδικός συνδυασμός μεταξύ του MMAS και του ACS: κατά την διάρκεια υλοποίησης της λύσης, τα μυρμήγκια στον ACS κάνουν την καλύτερη δυνατή επιλογή, όπως υποδεικνύεται από το ίχνος φερομόνης και τις ευρεστικές πληροφορίες με μια σταθερή πιθανότητα P και με την πιθανότητα $1-P$ επιλέγουν τον πιθανοκρατικό κανόνα της εξίσωσης (4.7). Με υψηλές τιμές της παραμέτρου P και το γεγονός ότι μόνο η καλύτερη λύση σε κάθε επανάληψη ή η συνολική βέλτιστη λύση επιλέγεται για την ανανέωση των ιχνών φερομόνης, προκύπτει μια ισχυρή αξιοποίηση των προηγούμενων αποτελεσμάτων αναζήτησης. Ένα κοινό χαρακτηριστικό γνώρισμα του MMAS με τους άλλους βελτιωμένους AS αλγορίθμους είναι το γεγονός ότι οι καλύτερες λύσεις που βρίσκονται κατά την διάρκεια της αναζήτησης αξιοποιούνται έντονα για να κατευθύνουν την αναζήτηση των μυρμηγκιών. Ένας υβριδικός συνδυασμός του MMAS και του ACS αποτελεί ένα ισχυρό

αλγοριθμικό εργαλείο για την επίλυση προβλήματων συνδυαστικής βελτιστοποίησης.

Στους αλγόριθμους ACO σημαντική σημασία έχει ο καθορισμός των παραμέτρων που τους συνθέτουν. Η έρευνα έδειξε ότι οι παράμετροι που δόθηκαν στα προβλήματα του Πίνακα 4-2 έχουν πολλή καλή απόδοση σε πάρα πολλά προβλήματα. Όμως, σε άλλες εφαρμογές, (προσαρμόσιμες εκδόσεις), οι οποίες συντονίζουν τις παραμέτρους κατά την διάρκεια εκτέλεσης του αλγορίθμου, μπορούν να αυξήσουν την ευρωστία του αλγορίθμου.

ΚΕΦΑΛΑΙΟ 5

ΘΕΩΡΗΤΙΚΗ ΘΕΜΕΛΙΩΣΗ ΤΩΝ ACO ΣΕ ΔΙΑΚΡΙΤΑ ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

5.1 Εισαγωγή

Η σύντομη ιστορία των μετα-ευρεστικών αλγορίθμων βελτιστοποίησης αποικίας μυρμηγκιών (ant colony optimization meta-heuristic) είναι κυρίως ιστορία πειραματικής έρευνας. Η δοκιμή και το λάθος αποτέλεσαν τον οδηγό για τους προγενέστερους ερευνητές καθώς επίσης και για τις περισσότερες τωρινές ερευνητικές προσπάθειες. Η παρακάτω θέση χαρακτηρίζει τυπικά όλους τους υπάρχοντες μετα-ευρεστικούς αλγορίθμους: «Το πρακτικό ενδιαφέρον για τους μετα-ευρεστικούς αλγορίθμους προήλθε μέσα από πειραματική εργασία, κατά την οποία οι ερευνητές προσπάθησαν να εμβαθύνουν το ενδιαφέρον τους σ' αυτούς κάνοντας, όχι μόνο, ολοένα και περισσότερα ειδικά πειράματα αλλά προσπαθώντας κυρίως να δημιουργήσουν μια θεωρία γι' αυτούς» [Dorigo και Stützle, 2004].

Σε αυτό το κεφάλαιο θα περιγραφεί η θεωρητική θεμελίωση των ACO σε διακριτά προβλήματα βελτιστοποίησης που λύνουν οι αλγόριθμοι αποικίας μυρμηγκιών.

Κατά την εφαρμογή των αλγορίθμων ACO σημαντική σημασία έχει η εξάτμιση της φερομόνης, ουσίας μέσω της οποίας επικοινωνούν τα μυρμήγκια μιας δοσμένης αποικίας. Εδώ δίνεται μια νέα μαθηματική ανάλυση και προσέγγιση του ρόλου της εξάτμισης της φερομόνης [Foundas και Vlachos, 2005], ενώ εξηγούνται οι έννοιες της γεωμετρικής και αλγεβρικής της σύγκλισης.

5.2 Θεωρητική περιγραφή των προβλημάτων εφαρμογής των ACO

Οι αλγόριθμοι ACO μπορούν να εφαρμοστούν σε διακριτά προβλήματα βελτιστοποίησης, που ορίζονται ως εξής:

- Δίνεται ένα πεπερασμένο σύνολο στοιχείων $C=\{c_1, c_2 \mathbf{K}, c_N\}$ το οποίο αποτελείται από c_i *συνιστώσες* ή *κόμβους*.

- Δίνεται ένα πεπερασμένο σύνολο L πιθανών συνδέσεων, $l_{c_i c_j}$ ή μεταβάσεων ανάμεσα στα στοιχεία του συνόλου C, ορισμένο σε ένα υποσύνολο $\hat{C}$ του καρτεσιανού γινομένου

$$C \times C, \quad L = \left\{ l_{c_i c_j} : (c_i c_j) \in \hat{C} \right\}, \quad |L| \le N_c^2$$

- Για κάθε σύνδεση ή τόξο $l_{c_i c_j} \in L$ ορίζεται μία συνάρτηση κόστους $J\left(l_{c_i c_j}, t\right)$, όπου t η παράμετρος του χρόνου διάσχισης του αντίστοιχου τόξου.

- Προσδιορίζεται ένα πεπερασμένο σύνολο περιορισμών $\Omega(C, L, t)$ από τα στοιχεία των συνόλων C και L.

- *Οι θέσεις* (states) του προβλήματος προσδιορίζονται από τους όρους της ακολουθίας $S=\{c_i, c_j, \mathbf{K}, c_k, \mathbf{K}\}$, η οποία ανήκει στο σύνολο C. Η ακολουθία περιγράφει τους κόμβους που επισκέπτεται το μυρμήγκι με τη σειρά που δείχνονται στην ακολουθία. *Η θέση* του προβλήματος αντιστοιχεί στη μνήμη του μυρμηγκιού κατά τη διάρκεια της διαδρομής του.

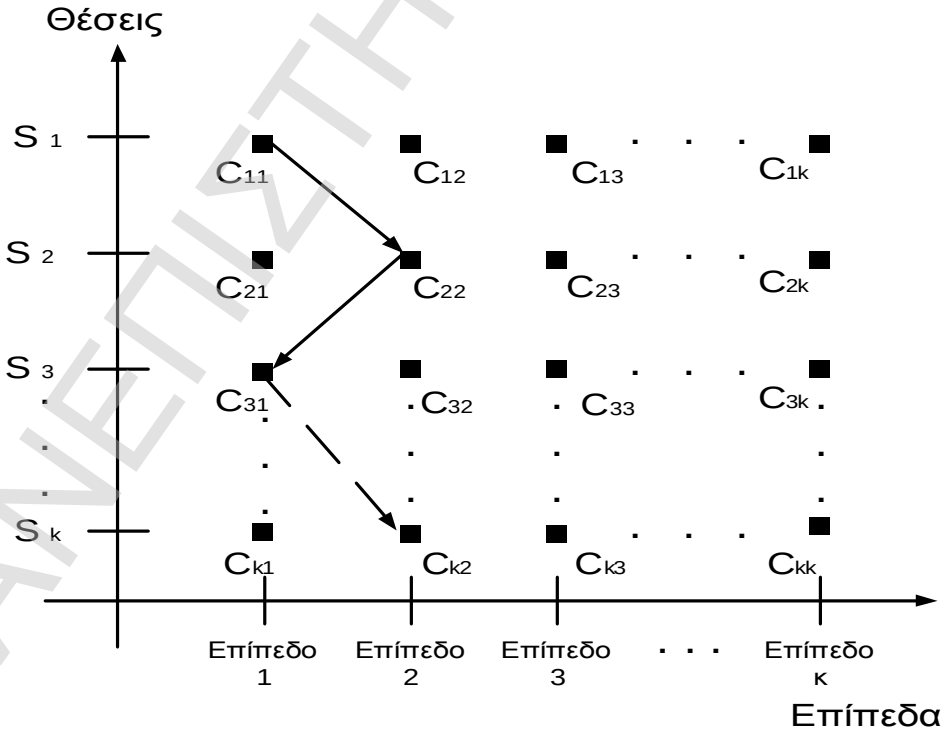
- Αν S είναι το σύνολο των δυνατών ακολουθιών, τότε $\hat{S}(t)$ είναι το σύνολο όλων των υπο-ακολουθιών που ικανοποιούν τους περιορισμούς: $\Omega(C, L, t)$. Τα στοιχεία του $\hat{S}(t)$ εκπροσωπούν τις εφικτές θέσεις (feasible states) του προβλήματος. Το μήκος της ακολουθίας S εκφράζεται ως $|S|$.

- Έστω δύο θέσεις $S_1 = \{c_i, c_j, \mathbf{K}, c_k\}$ και $S_2 = \{c_i, c_j, \mathbf{K}, c_k, c_{k+1}\}$ τότε η θέση S_2 είναι γειτονιά (neighborhood) της αντίστοιχης S_1, αν υπάρχει ένα $l_{c_k c_{k+1}} \in L$ έτσι ώστε να προκύπτει $S_2 = \{S_1, c_{k+1}\}$. Το σύνολο των δυνατών θέσεων στην S_i εκφράζεται ως N_{S_i}.

- Το σύνολο $\hat{N}_{S_i}(t) \subseteq N_{S_i}$ ονομάζεται σύνολο των γειτονικών δυνατών θέσεων και ορίζεται ως εξής: $\hat{N}_{S_i}(t) = N_{S_i} \cap \hat{S}(t)$.

Για την μελέτη ενός προβλήματος συνδυαστικής βελτιστοποίησης πρέπει να δημιουργηθεί ένα γράφημα δομής (construction graph) $G = (C, L)$, [Alonso και άλλοι, 2004], στο οποίο:

- οι συνιστώσες c_i αντιστοιχούν στους κόμβους του γραφήματος,
- οι θέσεις s αντιστοιχούν στις διαδρομές (τόξα) του γραφήματος, για παράδειγμα, στις ακολουθίες των κόμβων ή των τόξων,
- τα τόξα του γραφήματος, $l_{c_i c_j}$, συνδέονται έτσι ώστε να ορίζουν την αρχιτεκτονική της γειτονιάς (Σχήμα 5-1). $S^* = \{S, C\}$ είναι μια γειτονιά του S^* αν ο κόμβος c_i είναι η τελευταία συνιστώσα του S^* και το τόξο $l_{c_i c_j}$ υπάρχει στο γράφημα.



Σχήμα 5-1: Τυπικός ορισμός μιας γειτονιάς σε ένα γράφημα.

Ένα σύνολο που αποτελείται από τα στοιχεία $c_{11}, c_{22}, c_{31}, \mathbf{K}, c_{k2}$ είναι μια γειτονιά.

- υπάρχουν σαφείς συναρτήσεις κόστους που συνδέονται με κάθε τόξο,
- οι συνιστώσες και οι διαδρομές (τόξα) συνδέονται με ίχνη φερομόνης (τ), η οποία εκφράζει τον έμμεσο τρόπο επικοινωνίας καθώς και τη μνήμη των μυρμηγκιών στη διαδικασία της αναζήτησης καθώς και με τιμές ευρεστικών πληροφοριών (η) που εξαρτώνται από τη φύση του προβλήματος.

5.3 Θεμελιώδη στοιχεία των αλγορίθμων ACO

Οι αλγόριθμοι ACO είναι κατασκευαστικοί αλγόριθμοι: σε κάθε αλγοριθμική επανάληψη, κάθε μυρμήγκι κατασκευάζει μια λύση ενός προβλήματος, διασχίζοντας το γράφημα δομής. Σε κάθε τόξο του γραφήματος εντοπίζονται δύο ειδών πληροφορίες που καθοδηγούν την κίνηση του μυρμηγκιού:

i. *Μνημονική πληροφορία* τ_{ij}: αυτό το είδος της πληροφορίας αλλάζει στη διάρκεια της λειτουργίας του αλγορίθμου. Εξαρτάται από δύο παράγοντες: από τον αριθμό των μυρμηγκιών που διασχίζουν το τόξο και από την καλής ποιότητας διαδρομή (βέλτιστη λύση) που βρίσκουν τα αντίστοιχα μυρμήγκια. Σε σχέση με τα πραγματικά μυρμήγκια, η μνημονική πληροφορία αντιστοιχεί στη φερομόνη, χημική ουσία, που αφήνουν στο έδαφος. Συνήθως, ονομάζεται ίχνος φερομόνης.

ii. *Ευρεστική πληροφορία* η_{ij}: εξαρτάται μόνον από τη φύση του προβλήματος. Συνήθως, αυτή η πληροφορία υπολογίζεται πριν αρχίσει να λειτουργεί ο αλγόριθμος. Εκφράζει την a priori ποιότητα του τόξου. Σε σχέση με τα πραγματικά μυρμήγκια, η ευρεστική πληροφορία αντιστοιχεί στη δυσκολία διάσχισης ή την απόσταση ενός μέρους της διαδρομής τους. Εξαρτάται από την τοπογραφική δομή του εδάφους.

Εκμεταλλευόμενα τις ποσότητες φερομόνης και την ευρεστική πληροφορία, τα τεχνητά μυρμήγκια προσπαθούν να χτίσουν τις δυνατές λύσεις του προβλήματος. Τα τεχνητά μυρμήγκια έχουν τις παρακάτω ιδιότητες [Dorigo και Di Caro, 1999], [Dorigo και Stützle, 2003]:

- Αναζητούν τη λύση ελαχίστου κόστους min $\left(J_{\psi}(L,T)\right)$

- Κάθε μυρμήγκι k έχει μία μνήμη M^k που χρησιμοποιεί για να απομνημονεύσει πληροφορίες σχετικά με την διαδρομή του. Αυτές οι πληροφορίες μπορούν να χρησιμεύσουν:

  i. Για την εκτίμηση της καταλληλότητας της λύσης του προβλήματος, και

  ii. Για την επανάληψη της ήδη πραγματοποιηθείσης διαδρομής.

- Ένα μυρμήγκι ευρισκόμενο σε μία θέση S_i μπορεί να κινηθεί στην κατεύθυνση μιας άλλης θέσης S_{i+1} αν αυτή περιλαμβάνεται στο σύνολο των γειτονικών θέσεων $\hat{N}_i^k$ [5].

- Για κάθε θέση S_i ορίζεται η ορατότητα η_{ij} $\left(\eta_{ij} \in \hat{N}_i^k\right)$ σαν μέτρο της συμφέρουσας επιλογής για τη μετακίνηση από τη θέση S_i στη θέση, S_j. Σε αυτή την περίπτωση η συμφέρουσα επιλογή πρέπει να βασίζεται σε μία a priori πληροφορία.

- Κάθε μυρμήγκι k μπορεί να έχει μια αρχική θέση S_s^k, ενώ να έχει μία ή περισσότερες συνθήκες θανάτου e^k.

- Κάθε μυρμήγκι έχει την αφετηρία του σε μία αρχική θέση και κινείται στην κατεύθυνση των γειτονικών δυνατών θέσεων, δημιουργώντας τη λύση του προβλήματος βήμα-βήμα. Η διαδικασία σταματά όταν, τουλάχιστον, ένα μυρμήγκι ικανοποιεί τις συνθήκες θανάτου.

- Η κίνηση του μυρμηγκιού k από την θέση S_i στη θέση $S_j \in \hat{N}_i^k$ γίνεται με πιθανοκρατικό τρόπο που βρίσκεται σε κάθε κόμβο με τη μορφή της ant routing table $A_i = \left[\alpha_{ij}\right]$. Η ant routing table είναι συνάρτηση της φερομόνης $\tau_{ij}(t)$, της ευρεστικής τιμής η_{ij}, της διαδρομής που το μυρμήγκι έκανε μέχρι να επισκεφθεί αυτόν τον κόμβο $\left(M^k\right)$ και των περιοριστικών συνθηκών.

[5] Τα σύμβολα $\hat{N}_i^k$ και $\hat{N}_i^k(t)$ θα χρησιμοποιούνται με τον ίδιο τρόπο. Θεωρείστε ότι η παράμετρος t, έχει σημαντική σημασία, γιατί στην πράξη εκφράζει το χρόνο, επομένως τη δυνατότητα των μυρμηγκιών να διαφοροποιήσουν την αντίληψής τους όσο αφορά το πρόβλημα κατά τη διάρκεια ανατίμησής του.

- Όταν υλοποιηθεί μια λύση, τα μυρμήγκια μπορούν να ξανά-διανύσουν την ακολουθία των κόμβων προς τα πίσω και να εναποτεθεί εκ νέου φερομόνη. Αυτή η διαδικασία ονομάζεται: online delayed pheromone update.

Εκτός από τη δραστηριότητα του κάθε μυρμηγκιού, ξεχωριστά, υπάρχουν και κάποιες σημαντικές λειτουργίες που πρέπει να παίρνονται υπόψη :

i. η εξάτμιση της φερομόνης (pheromone evaporation) και

ii. η daemon ενέργεια (daemon action).

Αναλυτικά, η εξάτμιση της φερομόνης (η σημασία της δείχνεται στην παράγραφο 5.5) είναι μια διαδικασία κατά την οποία η πυκνότητα του ίχνους φερομόνης στα τόξα που συνδέουν τους κόμβους του γραφήματος μειώνεται, αυτόματα, στο χρόνο. Αυτό είναι αναγκαίο γιατί έτσι αποκλείεται η γρήγορη σύγκλιση του αλγορίθμου σε λύσεις που δεν είναι βέλτιστες, δίνοντας τη δυνατότητα στα μυρμήγκια να εξερευνήσουν άλλες περιοχές του γραφήματος. Οι deamon actions χρησιμοποιούνται στην εφαρμογή συγκεντρωτικών ενεργειών που δεν μπορούν να εκτελεστούν από μεμονωμένα μυρμήγκια. Παραδείγματα των deamon actions [Corne και άλλοι, 1999] είναι η ενεργοποίηση μιας διαδικασίας τοπικής βελτιστοποίησης, ή συλλογής συνολικών πληροφοριών που χρησιμεύουν στην εξαγωγή μιας απόφασης για το αν είναι χρήσιμη ή όχι η εναπόθεση επιπλέον φερομόνης, με σκοπό να απομακρυνθεί η ερευνητική διαδικασία από κάθε μη-τοπική περιοχή. Σε πρακτικό παράδειγμα, οι deamon actions μπορούν να παρατηρούν τη διαδρομή που εντοπίζει κάθε μυρμήγκι της αποικίας και να επιλέγουν την εναπόθεση επιπλέον φερομόνης στα τόξα που χρησιμοποιήθηκαν από το μυρμήγκι που διένυσε τη συντομότερη διαδρομή. Οι αναπροσαρμογές φερομονών που γίνονται με αυτόν ή παρόμοιους τρόπους από τις deamon actions ονομάζονται off line pheromone updates.

Στο Σχήμα 5-2 περιγράφεται με ψευδοκώδικα η συμπεριφορά των μετα-ευρεστικών αλγορίθμων βελτιστοποίησης.

```
1   procedure μετα-ευρεστικός_αλγόριθμος_ACO
2         while ( δεν_ικανοποιήθηκε_κριτηρίου_τερματισμού )
3               δραστηριότητες_προγράμματα
4                     δημιουργία_και_δραστηριότητα_μυρμηγκιών( );
5                     εξάτμιση_φερομόνης( );
6                     daemon _ενέργειες ( );              {προαιρετικό}
7               end δραστηριότητες_προγράμματα
8         end while
9   end procedure

1   procedure δημιουργία_και_δραστηριότητα_μυρμηγκιών( );
2         while διαθέσιμες_πηγές( );
3                     δημιουργία_ενός_νέου_μυρμηγκιού( );
4                     νέο_ενεργό_μυρμήγκι( );
5         end while
6   end procedure

1   procedure νέο_ενεργό_μυρμήγκι( ) {κύκλος ζωής μυρμηγκιού}
2           αρχικοποίηση_μυρμηγκιού( );
3           M = ανανέωση_μνήμης_μυρμηγκιού( );
4       while ( παρούσα_θέση ≠ θέσης_στόχου )
5           A = ανάγνωση_τοπικού_πίνακα_δρομολόγησης_μυρ/κιού( );
6           P = υπολογισμός_πιθανοτήτων_μετάβασης (A, M, Ω);
7           επόμενη_θέση = αποτέλεσμα_πολιτικής_απόφασης (P, Ω);
8           μετακίνηση_στην_επόμενη_πόλη(επόμενη_πόλη);
              if (online_βήμα_προς_βήμα_ανανέωση_φερομόνης)
9                 εναπόθεση_φερομόνης_στο_επισκεπτόμενο_τόξο( );
10                ανανέωση_πίνακα_δρομολόγησης_μυρμηγκιού ( );
              end if
11            M = ανανέωση_εσωτερικής_θέσης( );
12      end while
          if (online_delayed_ανανέωση_φερομόνης)
13          foreach επισκεπτόμενο_τόξο ∈ ψ do
14                εναπόθεση_φερομόνης_στο_επισκεπτόμενο_τόξο( );
15                ανανέωση_πίνακα_δρομολόγησης_μυρμηγκιού( );
16          end foreach
          end if
17        πεθαίνει( );
18  end procedure
```

Σχήμα 5-2: Ψευδοκώδικας των αλγορίθμων ACO [Corne και άλλοι, 1999].

Η κύρια διαδικασία των αλγορίθμων ACO είναι να καταφέρνει, μέσω της κατασκευής των δραστηριοτήτων του προγράμματος, να σχεδιάζει τρία βασικά συστατικά τους:

i. Τη δημιουργία και δραστηριότητα των μυρμηγκιών,

ii. Την εξάτμιση της φερομόνης και

iii. Τις daemon actions.

Στο νέο ενεργό μυρμήγκι ( ), ψευδοκώδικας Σχ. 5-2, η θέση-στόχος ($4^{η}$ γραμμή) αναφέρεται σε μία ολοκληρωμένη λύση που υλοποιείται από το μυρμήγκι. Οι διαδικασίες βήμα προς βήμα και delayed ανανέωση φερομόνης, στις γραμμές 9-10 και 14-15 είναι, συχνά, αμοιβαία αποκλειόμενες. Όταν και οι δύο είναι απούσες, η φερομόνη εναποτίθεται από τις daemon actions.

5.4 Βήματα για τη λύση προβλημάτων που χρησιμοποιούν αλγορίθμους ACO

Από τις ήδη γνωστές εφαρμογές της ACO, μπορούν να προσδιοριστούν μερικές οδηγίες για την επίλυση νέων προβλημάτων με τη βοήθεια των αλγορίθμων ACO. Οι οδηγίες αυτές μπορούν να συνοψιστούν στα παρακάτω έξι σχεδιαστικά βήματα [Cordon, Herrera και Stützle, 2002]:

i. Αναπαράσταση του προβλήματος με τη μορφή συνόλων από συνιστώσες και μεταβάσεις ή υπό την μορφή γραφήματος το οποίο το διασχίζουν τα μυρμήγκια προκειμένου να σχηματίσουν λύσεις.

ii. Κατάλληλος ορισμός του νοήματος του μονοπατιού φερομόνης (ίχνη φερομόνης) τ_{ij} (για παράδειγμα, το είδος της απόφασης που προκαταλαμβάνουν). Αυτό είναι ένα πολύ σημαντικό βήμα στην υλοποίηση ενός αλγορίθμου ACO και συχνά ένας καλός ορισμός του μονοπατιού φερομόνης δεν είναι μια ασήμαντη εργασία αλλά απαιτεί βαθιά επίγνωση του προβλήματος προς επίλυση.

iii. Κατάλληλος ορισμός της ευρεστικής προτίμησης σε κάθε απόφαση που πρέπει ένα μυρμήγκι να πάρει κατά τον σχηματισμό μιας λύσης (για παράδειγμα, ο ορισμός της ευρεστικής πληροφορίας η_{ij} που συσχετίζεται με κάθε συνιστώσα ή μετάβαση). Σημειώνεται, ότι η

ευρεστική πληροφορία είναι σημαντική για καλή απόδοση σε περίπτωση που οι αλγόριθμοι τοπικής αναζήτησης δεν είναι διαθέσιμοι ή δεν μπορούν να εφαρμοστούν.

iv. Αν είναι εφικτό, υλοποίηση ενός αποτελεσματικού αλγορίθμου τοπικής αναζήτησης για το πρόβλημα προς επίλυση, επειδή τα αποτελέσματα πολλών ACO εφαρμογών σε NP – «δύσκολα» συνδυαστικά προβλήματα βελτιστοποίησης δείχνουν ότι η βέλτιστη απόδοση επιτυγχάνεται όταν συνδυάζονται οι αλγόριθμοι ACO με αλγόριθμους τοπικής αναζήτησης [Dorigo και Di Caro, 1999], [Dorigo και Stützle, 2003].

v. Επιλογή ενός συγκεκριμένου αλγορίθμου ACO και εφαρμογή του στο πρόβλημα προς επίλυση, λαμβάνοντας υπόψη τους παραπάνω κανόνες.

vi. Ρύθμιση των παραμέτρων του αλγορίθμου ACO. Μια καλή αρχή για την ρύθμιση των παραμέτρων είναι να χρησιμοποιηθούν οι ρυθμίσεις των παραμέτρων που αποδείχτηκαν καλές κατά την εφαρμογή του αλγορίθμου ACO σε παρόμοια προβλήματα. Μια εναλλακτική λύση της, χρονοβόρας για τον άνθρωπο, ρύθμισης των παραμέτρων είναι η χρήση αυτόματων διαδικασιών που θα ρυθμίζουν τις παραμέτρους [Birattari και άλλοι, 2002].

Ευνόητο είναι ότι τα παραπάνω βήματα μπορούν να δώσουν μόνο μια πρόχειρη κατεύθυνση στην υλοποίηση των αλγορίθμων ACO. Επιπρόσθετα, επειδή η υλοποίηση είναι συνήθως μια επαναληπτική διαδικασία, θα πρέπει, εμβαθύνοντας περισσότερο στο πρόβλημα και στη συμπεριφορά του αλγορίθμου, να αναθεωρηθούν μερικές αρχικές επιλογές. Τέλος, σημειώνεται ότι τα περισσότερο πιο σημαντικά βήματα είναι μάλλον τα τέσσερα πρώτα, επειδή μια κακή επιλογή σε αυτό το στάδιο δεν μπορεί να αναπληρωθεί με μια καλή ρύθμιση των παραμέτρων.

5.5 Μια νέα μαθηματική προσέγγιση της εξάτμισης της φερομόνης των αλγορίθμων ACO

Σημειώθηκε, στις παραπάνω παραγράφους ο σημαντικός ρόλος της εξάτμισης φερομονών, σχετικά με τη σύγκλιση των αλγορίθμων ACO. Σε αυτή την παράγραφο θεμελιώνεται το μαθηματικό μοντέλο της εξάτμισης

της φερομόνης [Foundas και Vlachos, 2005], ενώ στη συνέχεια εξετάζεται η γεωμετρική και η αλγεβρική της σύγκλιση.

Για την εξάτμιση της φερομόνης, τη χρονική στιγμή t, ρ_t, είναι γνωστό ότι ικανοποιεί τη συνθήκη $0 \le \rho_t \le 1$ [Bullnheimer Bernd και άλλοι, 1998]. Για τη χρονική στιγμή $t+1$, η ρ_{t+1} γράφεται:

$$\rho_{t+1} \equiv \rho_t + \alpha(1-\rho_t) - \beta\rho_t \qquad (5.1)$$

με

$$0 \le \rho_t \le 1, \quad 0 \le \alpha \le 1, \quad 0 \le \beta \le 1$$

όπου α και β σταθερές.

Θα δειχθεί ότι $0 \le \rho_{t+1} \le 1$. (5.2)

Απόδειξη

Ισχύει ότι :

$$\rho_{t+1} = \rho_t + \alpha(1-\rho_t) - \beta\rho_t \le \rho_t + 1(1-\rho_t) + 0\rho_t = 1$$

και

$$\rho_{t+1} = \rho_t + \alpha(1-\rho_t) - \beta\rho_t \ge \rho_t + 0(1-\rho_t) - 1\rho_t = 0$$

οπότε ισχύει η σχέση (5.2).

Παίρνοντας υπόψη τη σχέση (5.2), μπορεί να γραφτεί η εναλλακτική μορφή της εξίσωσης (5.1), ως εξής:

$$\rho_t = \kappa\rho_{t-1} + \alpha \qquad (5.3)$$

Όπου κ η *ταχύτητα εξάτμισης* της φερομόνης για την οποία ισχύει:

$$\kappa = 1-\alpha-\beta \text{ και } 0 < \kappa < 1$$

Η γεννήτρια συνάρτησης της εξίσωσης (5.3) είναι:

$$f(x)=\sum_{t=0}^{\infty}\rho_t x^t$$

και

$$\begin{aligned} f(x) &= \sum_{t=1}^{\infty}(\kappa\rho_{t-1}+\alpha)x^t \\ &= \kappa\sum_{t=1}^{\infty}\rho_{t-1}x^t+\alpha\sum_{t=0}^{\infty}x^t \\ &= \kappa\sum_{t=0}^{\infty}\rho_t x^{t+1}+\alpha\frac{1}{1-x} \end{aligned}$$

και

$$\begin{aligned} f(x) &= \kappa x\sum_{t=0}^{\infty}\rho_t x^t+\alpha\frac{1}{1-x} \\ &= \kappa x f(x)+\frac{\alpha}{1-x} \end{aligned}$$

και

$$f(x)=\frac{\alpha}{(1-\kappa x)(1-x)}$$

Η συνάρτηση $f(x)$ γράφεται:

$$\frac{\alpha}{(1-\kappa x)(1-x)}\equiv\frac{A}{1-\kappa x}+\frac{B}{1-x}$$

οπότε:

$$A=\frac{\alpha\kappa}{\kappa-1},\quad B=\frac{\alpha}{1-\kappa}$$

και

$$f(x) \equiv \frac{\alpha\kappa}{\kappa-1}\frac{1}{1-\kappa x}+\frac{\alpha}{1-\kappa}\frac{1}{1-x}$$
$$=\frac{\alpha\kappa}{\kappa-1}\sum_{t=0}^{\infty}(\kappa x)^t+\frac{\alpha}{1-\kappa}\sum_{t=0}^{\infty}x^t$$

Άρα

$$f(x) \equiv \sum_{t=0}^{\infty}\left(\frac{\alpha\kappa}{\kappa-1}\kappa^t+\frac{\alpha}{1-\kappa}\right)x^t$$

και

$$\rho_t=\frac{\alpha\kappa}{\kappa-1}\kappa^t+\frac{\alpha}{1-\kappa}$$

οπότε:

$$\rho_t \equiv \begin{cases}\left(\rho_0-\frac{\alpha}{1-\kappa}\right)\kappa^t+\frac{\alpha}{1-\kappa} & , \quad \alpha\nu \quad \kappa\neq 1 \\ \rho_0 & , \quad \alpha\nu \quad \kappa=1\end{cases} \qquad (5.4)$$

Αν $C=\left(\rho_0-\frac{\alpha}{1-\kappa}\right)$ και $0\le t\le t_0$, θα δειχθεί ότι η εξίσωση (5.4) είναι μια *συνάρτηση πυκνότητας πιθανότητα* :

$$\rho_t=\int_0^{t_0}\left(c\kappa^t+\frac{\alpha}{1-\kappa}\right)dt=1$$

είναι

$$C=\left(1-\frac{\alpha t_0}{1-\kappa}\right)\left(\frac{\ln\kappa}{\kappa^{t_0}-1}\right)$$

Ισχύει:

$$\int_0^{t_0}\left(1-\frac{\alpha t_0}{1-\kappa}\right)\left(\frac{\ln\kappa}{\kappa^{t_0}-1}\right)\kappa^t dt+\frac{\alpha t}{1-\kappa}\Big]_0^{t_0}=$$

$$\left(1-\frac{\alpha t_0}{1-\kappa}\right)\left(\frac{\ln\kappa}{\kappa^{t_0}-1}\right)\frac{\kappa^t}{\lim\kappa}\Big]_0^{t_0}-\left(1-\frac{\alpha t_0}{1-\kappa}\right)\left(\frac{1}{\kappa^{t_0}-1}\right)+\frac{\alpha t_0}{1-\kappa}=1$$

γιατί

$$\left(1-\frac{\alpha t_0}{1-\kappa}\right)\left(\frac{\kappa^{t_0}}{\kappa^{t_0}}-\frac{1}{\kappa^{t_0}-1}\right)+\frac{\alpha t_0}{1-\kappa}=1$$

εφόσον

$$\left(\frac{\kappa^{t_0}}{\kappa^{t_0}-1}-\frac{1}{\kappa^{t_0}-1}\right)=1$$

άρα η εξίσωση (5.4) είναι μια *συνάρτηση πυκνότητας πιθανότητας*.

5.5-1 Γεωμετρική ερμηνεία της σύγκλισης

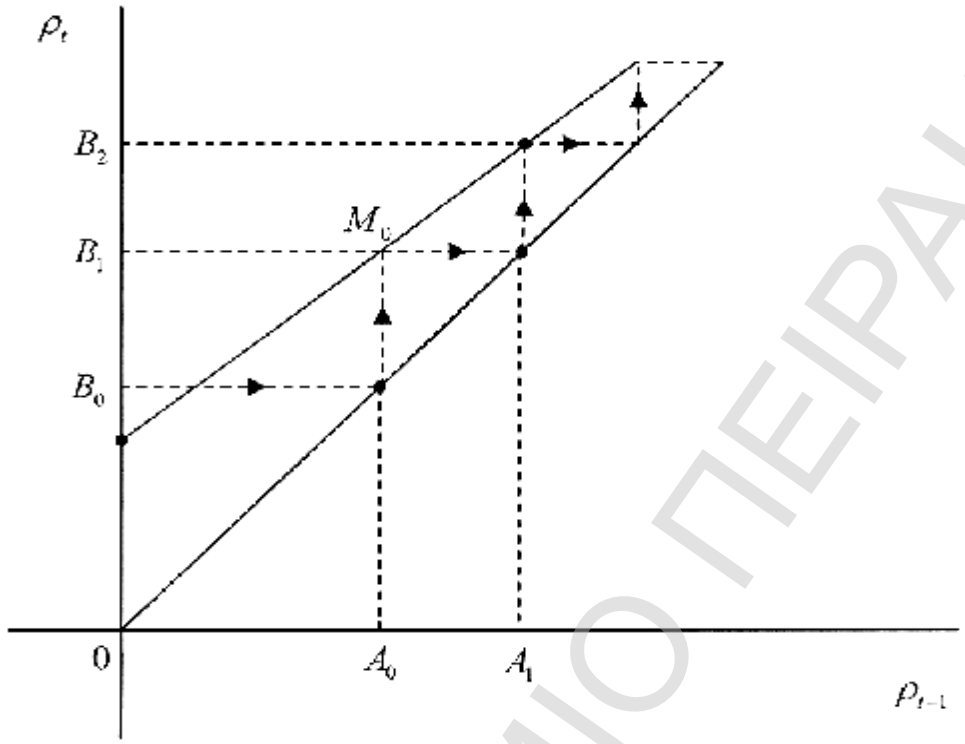
Από την εξίσωση:

$$\rho_t=\kappa\rho_{t-1}+\alpha,\quad 0<\kappa<1\quad \text{και}\quad 0\le\alpha\le 1$$

προκύπτει

$$\lim_{t\to\infty}=\begin{cases}\dfrac{\alpha}{1-\kappa}, & \text{αν}\quad 0<\kappa<1,\quad \alpha\neq 0\\ 0, & \text{αν}\quad 0<\kappa<1,\quad \alpha=0\end{cases}$$

η σύγκλιση της εξίσωσης (5.4) είναι η τιμή $\frac{\alpha}{1-k}$, όταν $0<\kappa<1$ και $\alpha\neq 0$, (Σχήμα 5-3).



Σχήμα 5-3: Διαγραμματική ερμηνεία της σύγκλισης.

Το σημείο $B_1(0,\rho_1)$ αντιστοιχεί στην εξίσωση $\rho_1 = \kappa t_0 + \alpha$, ενώ το σημείο $B_2(0,\rho_2)$ αντιστοιχεί στην εξίσωση $\rho_2 = \kappa\rho_1 + \alpha$. Συνεχίζοντας τη διαδικασία, τελικά, το σημείο τομής $T\left(\frac{\alpha}{1-\kappa}, \frac{\alpha}{1-\kappa}\right)$ της διχοτόμου $\rho_t = \rho_{t-1}$ και της ευθείας $\rho_t = \kappa\rho_{t-1} + \alpha$, είναι η σύγκλιση της ακολουθίας.

5.5-2 Αλγεβρική ερμηνεία της σύγκλισης

Από την εξίσωση (5.4) προκύπτει, για $t = 1,2,3,\mathbf{K},t$:

$$\rho_1 = \kappa\rho_0 + \alpha$$

$$\rho_2 = \kappa^2\rho_0 + (1+\kappa)\alpha$$

$$\rho_3 = \kappa^3\rho_0 + (1+\kappa+\kappa^2)\alpha$$

M M M

$$\rho_t = \kappa^t\rho_0 + (1+\kappa+\mathbf{L}+\kappa^{t+1})\alpha$$

και

$$\rho_t = \kappa^t\rho_0 + \frac{1-\kappa^t}{1-\kappa}\alpha$$

οπότε

$$\rho_t = \left(\rho_0 - \frac{\alpha}{1-\kappa}\right)\kappa^t + \frac{\alpha}{1-\kappa}$$

είναι:

$$\lim_{t\to\infty} = \begin{cases} \dfrac{\alpha}{1-\kappa}, & 0<\kappa<1, \quad \alpha\neq 0 \\ 0 \quad , & 0<\kappa<1, \quad \alpha = 0 \end{cases}$$

Κατά αυτόν τον τρόπο αποδείχτηκε η αλγεβρική σύγκλιση της ακολουθίας.

ΚΕΦΑΛΑΙΟ 6

ΜΟΝΤΕΛΑ ΔΙΑΧΕΙΡΙΣΗΣ ΦΕΡΟΜΟΝΗΣ

6.1 Εισαγωγή

Η μεγάλη αξία των αλγορίθμων αποικίας μυρμηγκιών στηρίζεται στη διαχείριση των χρησιμοποιούμενων ιχνών φερομόνης που σε συνδυασμό με την αντικειμενική συνάρτηση, ενός δεδομένου προβλήματος, κατασκευάζουν νέες λύσεις. Η πληροφορία που περιλαμβάνεται στα ίχνη φερομόνης και η χρήση της αποτελούν το κλειδί της καταξίωσης των αλγορίθμων ACO. Τα επίπεδα φερομόνης δίνουν το μέτρο του πόσο επιθυμητή είναι η εισαγωγή ενός δεδομένου στοιχείου σε μια λύση ενός προβλήματος. Τα ίχνη φερομόνης χρησιμοποιούνται για τη διαδικασία της εξερεύνησης (exploration) και της εκμετάλλευσης (exploitation) αντίστοιχα [Gambardella και Dorigo, 2000]. Η εξερεύνηση αφορά τον πιθανοκρατικό τρόπο επιλογής των συνιστωσών που χρησιμοποιούνται στην υλοποίηση μιας λύσης: τα στοιχεία με μεγάλη συγκέντρωση ιχνών φερομόνης, επιλέγονται με μεγάλη πιθανότητα. Στη διαδικασία της εκμετάλλευσης περιλαμβάνονται στοιχεία που μεγιστοποιούν ένα μίγμα τιμών ιχνών φερομόνης και μέρους των αντιστοίχων τιμών μιας αντικειμενικής συνάρτησης.

Στις επόμενες παραγράφους δίνονται διάφορα μοντέλα διαχείρισης φερομόνης και αντι-φερομόνης, που εφαρμόστηκαν στο TSP των M. Randall και J. Montgomery. Στη συνέχεια αποδεικνύεται το μαθηματικό μοντέλο της γενικευμένης μορφής διαχείρισης φερομόνης και δημιουργούνται μαθηματικά μοντέλα φερομόνης των αλγορίθμων ACS και MMAS μέσω των Εξισώσεων Διαφορών [Foundas και Vlachos, 2005].

6.2 Εναλλακτικοί τρόποι διαχείρισης φερομόνης στους αλγορίθμους ACO

Προτείνονται τρεις εναλλακτικοί τρόποι διαχείρισης της φερομόνης [Montgomery και Randall, 2002] στον αλγόριθμο Ant Colony System

(ACS) που χρησιμοποιείται στη λύση του προβλήματος του περιοδεύοντος πωλητή (TSP).

6.2-1 Τοποθέτηση φερομόνης στις πόλεις

Η πληροφορία της αποθηκευμένης φερομόνης για κάθε τόξο μπορεί να εξαρτάται έντονα από τη μνήμη, καθώς ο αριθμός των τόξων είναι $O(n^2)$, όπου n είναι ο αριθμός των πόλεων. Οι παρακάτω δύο τροποποιήσεις στην χρήση της φερομόνης, μείωσαν δραματικά τη μεγάλη χρήση της μνήμης, εναποθέτοντας φερομόνη στις πόλεις. Οι κανόνες μετάβασης του ACS είναι:

$$s = \begin{cases} \arg\max_{u \in N_r^k} \left\{ \tau_u(t) \left[d_{r,u} \right]^{\beta} \right\}, & \text{αν} \quad q \le q_0 \\ \text{Εξίσωση} \quad (6.2) \qquad , & \text{αλλιώς} \end{cases} \tag{6.1}$$

$$P_{r,s}^k = \begin{cases} \dfrac{\tau_s(t) \left[d_{r,s} \right]^{\beta}}{\sum_{u \in N_r^k} \tau_u(t) \left[d_{r,u} \right]^{\beta}}, & \text{αν} \quad s \in N_r^k \\ 0 \qquad\qquad\qquad\qquad , & \text{αλλιώς} \end{cases} \tag{6.2}$$

όπου:

$P_{r,s}^k$: ο πιθανοκρατικός κανόνας επιλογής του k-οστού μυρμηγκιού της πόλεως s όταν αυτό, αρχικά βρίσκεται στην πόλη r.

$q \in [0,1]$: είναι ένας σταθερός τυχαίος αριθμός και q_0 είναι μια παράμετρος. Για να διατηρήσει τον περιορισμό της μοναδικής επίσκεψης, το μυρμήγκι k εμποδίζεται να επιλέξει μια πόλη που έχει ήδη επισκεφτεί.

N_r^k: είναι οι πόλεις που δεν έχουν ακόμα επισκεφθεί από το μυρμήγκι k.

$d_{r,u}$: είναι η απόσταση μεταξύ των πόλεων r και u.

Δεδομένου της φύσης του προβλήματος του περιοδεύοντος πωλητή (TSP), κάθε μυρμήγκι πρέπει να επισκεφτεί κάθε πόλη, ακριβώς, μόνο μια φορά στη διάρκεια μιας επανάληψης και εναποθέτοντας την ίδια ποσότητα φερομόνης σε κάθε επίσκεψη δεν θα επιτύχει κάποια σημαντική αλλαγή στο ποσό της φερομόνης ανάμεσα στις πόλεις. Το κύριο χαρακτηριστικό των αλγορίθμων ACO είναι η μεταφορά ίσης πληροφορίας, φερομόνης, μεταξύ των επαναλήψεων και επομένως το γεγονός ότι υπάρχουν μικρές διαφορές στα επίπεδα φερομόνης σε μια επανάληψη, καθώς τα μυρμήγκια επισκέπτονται τις πόλεις με διαφορετική σειρά, αγνοείται. Προτείνονται δύο εναλλακτικοί τρόποι διαχείρισης φερομόνης, η οποία τοποθετείται στις πόλεις:

- *Phero City Perm* (PCP) και
- *Phero City Cost* (PCC).

Δεδομένου ενός προβλήματος TSP με n πόλεις, η λύση που παράγεται από το μυρμήγκι k, εκφράζεται από i_{ζ_k} πόλεις, με $i=[0,1,\mathbf{K},n-1]$. Στον PCP ένα μυρμήγκι εναποθέτει φερομόνη σε κάθε πόλη που επισκέπτεται, ανάλογα με το πόσο σύντομα η πόλη αυτή προστίθεται στην λύση του συγκεκριμένου μυρμηγκιού. Με αυτόν τον τρόπο, οι πόλεις που βρίσκονται στην αρχή της μετάθεσης των πόλεων, η οποία αποτελεί μια λύση, γίνονται πιο επιθυμητές από αυτές που βρίσκονται στο τέλος της. Ο κανόνας τοπικής ανανέωσης φερομόνης του ACS δίνεται:

$$\tau_s(t+1)=(1-\rho)\tau_s(t)+\rho\tau_0\frac{n\text{-}i}{n} \qquad (6.3)$$

όπου:

τ_0: είναι η αρχική ποσότητα φερομόνης που έχει εναποτεθεί σε κάθε πόλη

i: είναι η θέση στην οποία η πόλη s θα τοποθετηθεί κατά το τρέχον βήμα.

Ο κανόνας συνολικής ανανέωσης φερομόνης δίνεται:

$$\tau_s(t+1)=(1-\rho)\tau_s(t)+\rho\frac{Q}{L}\frac{n-\mathbf{i}_{\zeta^{k(s)}}}{n} \qquad (6.4)$$

όπου:

$\mathbf{i}_{\zeta^{k(s)}}$: είναι η θέση της πόλης s στην συνολική βέλτιστη λύση.

Ο PCC διαφέρει από τον PCP γιατί τα μυρμήγκια εναποθέτουν φερομόνη στις πόλεις ανάλογα με το κόστος του προστιθέμενου τόξου. Έτσι, τα μυρμήγκια που φθάνουν σε μια πόλη μέσω ενός «μεγάλου κόστους" τόξο εναποθέτουν λιγότερη φερομόνη σε αυτή την πόλη από τα μυρμήγκια που φθάνουν μέσω ενός συντομότερου («χαμηλού κόστους») τόξου. Για αυτό το λόγο, ο PCC χρησιμοποιεί τον ίδιο κανόνα επιλογής πόλης με τον PCP, αλλά χρησιμοποιεί διαφορετικούς κανόνες ανανέωσης φερομόνης. Ο κανόνας τοπικής ανανέωσης στον ACS είναι:

$$\tau_s(t+1) = (1-\rho)\tau_s(t) + \rho\tau_0\left[1 - \frac{d\left(\zeta^{\kappa}(i\text{-}1),s\right) - d_{min}}{d_{max} - d_{min}}\right] \qquad (6.5)$$

όπου:

$\zeta^{\kappa}(i-1)$: είναι η πόλη από την οποία το μυρμήγκι k μόλις έφθασε

d_{min} και d_{max}: είναι τα μήκη του μικρότερου και του μεγαλύτερου τόξου, στο πρόβλημα αντίστοιχα.

Ο κανόνας συνολικής ανανέωσης είναι:

$$\tau_s(t+1) = (1-\rho)\tau_s(t) + \rho\frac{Q}{L}\left[1 - \frac{d_{r,s} - d_{min}}{d_{max} - d_{min}}\right] \qquad (6.6)$$

με

$$r = \begin{cases} \zeta^{k}\left[\mathbf{i}_{\zeta^{k(s)}}\text{-}1\right] & , \quad \text{αν} \quad \mathbf{i}_{\zeta^{k(s)}} > 0 \\ \zeta^{k}(n\text{-}1) & , \quad \text{αλλιώς} \end{cases} \qquad (6.7)$$

όπου:

r: είναι η πόλη που προηγείται της πόλης s στην συνολική βέλτιστη λύση

L: είναι το μήκος της βέλτιστης (συντομότερης) διαδρομής

Q: είναι μια παράμετρος που εξαρτάται από το πρόβλημα.

6.2-2 Φερομόνη στη θέση της πόλης

Μια εναλλακτική προσέγγιση του παραδοσιακού τρόπου εναπόθεσης της φερομόνης στα τόξα, είναι η εναπόθεση φερομόνης *στη θέση* της πόλης, σε μια λύση. Αυτό το μοντέλο διαχείρισης σχετίζεται με την έννοια της προώθησης καλών μεταθέσεων των πόλεων. Η προσέγγιση αυτή αναφέρεται ως ***Phero Position*** (PP). Ένα μυρμήγκι k, που βρίσκεται στην πόλη r στην αρχή του βήματος i, επιλέγει να μετακινηθεί στην επόμενη πόλη s ως εξής:

$$s = \begin{cases} \arg\max_{u \in N_r^k} \left\{ \tau_{i,u}(t) \left[d_{r,u} \right]^{\beta} \right\}, & \text{αν} \quad q \le q_0 \\ \text{Εξίσωση} \quad (6.9) & , \quad \text{αλλιώς} \end{cases} \tag{6.8}$$

$$P^{k}_{r,i,s} = \begin{cases} \dfrac{\tau_{i,s}(t) \left[d_{r,s} \right]^{\beta}}{\sum_{u \in N_r^k} \tau_{i,u}(t) \left[d_{r,u} \right]^{\beta}}, & \text{αν} \quad s \in N_r^k \\ 0 & , \quad \text{αλλιώς} \end{cases} \tag{6.9}$$

όπου:

$P^{k}_{r,i,s}$: είναι η πιθανότητα με την οποία το μυρμήγκι k επιλέγει την πόλη s για την θέση i στην λύση του, δεδομένου ότι βρίσκεται στην πόλη r.

Οι κανόνες τοπικής και συνολικής ανανέωσης φερομόνης τροποποιούνται στην εναπόθεση φερομόνης *στη θέση* της πόλης, ως εξής:

$$\tau_{i,s}(t+1) = (1-\rho)\tau_{i,s}(t) + \rho\tau_0 \qquad (6.10)$$

και

$$\tau_{i,s}(t+1) = (1-\rho)\tau_{i,s}(t) + \rho\Delta\tau_{i,s}(t) \qquad (6.11)$$

με

$$\Delta\tau_{i,s}(t) = \begin{cases} \dfrac{Q}{L} & , \quad \text{αν} \quad \zeta^k = s \\ 0 & , \quad \text{αλλιώς} \end{cases}$$

όπου:

ζ^k: είναι η πόλη στην i-οστή θέση στην συνολική βέλτιστη λύση,

$\Delta\tau_{i,s}(t)$: χρησιμοποιείται για να ενισχύσει την φερομόνη στις θέσεις των πόλεων στην συνολική βέλτιστη λύση.

L: είναι το μήκος της βέλτιστης (συντομότερης) διαδρομής.

6.2-3 Αποτελέσματα εφαρμογής των αλγορίθμων PCC, PCP, PP

Εφαρμόζοντας τους τρεις τρόπους διαχείρισης της φερομόνης [Montgomery και Randall, 2002] σε διάφορους τύπους προβλημάτων, προέκυψαν τα αποτελέσματα που δείχνονται στον Πίνακα 6-1.

Τα ελάχιστο (minimum, “Min”), μέσο (median, “Med”), μέγιστο (maximum, “Max”) και inter-quartile range (“IQR”), Πίνακας 6-1, χρησιμοποιούνται για να συνοψίσουν τα αποτελέσματα καθώς αυτά δεν είναι ομοιόμορφα κατανεμημένα.

Ο πιο αποτελεσματικός από τους τρεις τρόπους διαχείρισης φερομόνης είναι ο PCC, στον οποίο περισσότερη ποσότητα φερομόνης εναποτίθεται στις επισκεπτόμενες πόλεις μέσω «χαμηλού κόστους» τόξων.

Τεχνική	Πρόβλημα	Κόστος			
		Min	Med	Max	IQR
Control	gr24	1272	1278	1328	1
	eil51	430	435	441	4
	eil76	546	558	564	5
	kroA100	21292	21393	22030	474
	d198	15954	16101	16253	209
	lin318	45716	47121	48144	1162
	pcb442	60970	63025	64858	1614
PheroCityPerm	gr24	1278	1345	1412	77
	eil51	459	474	487	6
	eil76	580	597	609	13
	kroA100	23688	24522	25172	518
	d198	17681	18095	18494	260
	lin318	50130	51736	52513	1155
	pcb442	63993	66195	67494	952
PheroCityCost	gr24	1272	1272	1278	6
	eil51	442	453	456	5
	eil76	567	576	580	6
	kroA100	23064	23576	23736	210
	d198	17264	17685	17911	196
	lin318	50069	50392	51184	549
	pcb442	65062	66886	67669	1438
PheroPosition	gr24	1283	1333	1444	100
	eil51	477	498	533	21
	eil76	597	639	677	30
	kroA100	23029	25008	26559	351
	d198	18211	18599	19569	382
	lin318	52736	53745	55425	656
	pcb442	66839	69664	70445	1344

Πίνακας 6-1: Αποτελέσματα εφαρμογής των τριών τρόπων διαχείρισης φερομόνης στον αλγόριθμο ACS.

6.3 Μοντέλο διαχείρισης στις Αντί-φερομόνης (Anti-pheromone)

Οι Schoonderwoerd και άλλοι (1996) ήταν οι πρώτοι που υπέδειξαν την χρήση του όρου «αντί-φερομόνη» (anti- pheromone), το αποτέλεσμα της οποίας είναι αντίθετο από την κανονική φερομόνη. Η αντί-φερομόνη είναι μια τεχνική των αλγορίθμων αποικίας μυρμηγκιών που χρησιμοποιείται για την λύση προβλημάτων συνδυαστικής βελτιστοποίησης και ειδικά στο TSP [Montgomery και Randall, 2002].

Η αντί-φερομόνη, ή οποιαδήποτε άλλη παραλλαγή του τύπου της φερομόνης που χρησιμοποιείται στους αλγόριθμους μυρμηγκιών, είναι απλά μια ουσία με διαφορετικό αποτέλεσμα από αυτό της «κανονικής» φερομόνης. Πράγματι, η αντί-φερομόνη είναι, γενικά, μια ουσία, που ευρισκόμενη στα στοιχεία των πιο φτωχών λύσεων, μπορεί να έχει παρόμοιο αποτέλεσμα, γνωστοποιώντας στις συσσωρευμένες κακές εμπειρίες των μυρμηγκιών που διαφορετικά θα είχαν χαθεί. Σε αυτή τη διατριβή θα χρησιμοποιηθούν τρεις αλγόριθμοι. Οι δυο πρώτοι αλγόριθμοι αφοράνε την διαχείριση της αντί-φερομόνης σαν μια μέθοδο καλύτερης εξερεύνησης του χώρου αναζήτησης. Ο τρίτος αλγόριθμος ακολουθεί μια διαφορετική προσέγγιση, κάνοντας την κανονική φερομόνη απωθητική σε ένα μικρό αριθμό μυρμηγκιών, από το να προσθέσει αντί-φερομόνη στις πιο φτωχές λύσεις. Σημειώνεται, ότι η φερομόνη που βλέπουν τα μυρμήγκια σαν απωθητική ουσία, εκφράζεται, για αυτά, σαν αντί-φερομόνη.

6.3-1 Αφαιρετική Αντί-φερομόνη (Subtractive Anti- pheromone, SAP)

Όταν τα μυρμήγκια δημιουργούν λύσεις, συχνά αναγνωρίζουν τόσο, σχετικά, φτωχές λύσεις όσο και καλές λύσεις. Σε αυτήν την έκδοση του αλγορίθμου ACS, ιδιαίτερη προσοχή δίνεται στις φτωχότερες λύσεις όπου η φερομόνη αφαιρείται από εκείνα τα στοιχεία που δημιουργούν την χείριστη λύση σε κάθε επανάληψη. Επομένως, οι επόμενες γενιές των μυρμηγκιών που θα ακολουθήσουν αποθαρρύνονται να χρησιμοποιήσουν στοιχεία που αποτελούσαν μέρος φτωχότερων λύσεων στο παρελθόν. Αυτό αποτελεί τον πιο απλό τρόπο εφαρμογής της αντί-φερομόνης όπου η εναπόθεση μιας απωθητικής φερομόνης προσομοιώνεται από μια μείωση της υπάρχουσας ποσότητας φερομόνης.

Ο αλγόριθμος SAP είναι ίδιος με τον ACS, εκτός από την προσθήκη ενός επιπλέον μέρους στην συνολική ανανέωση φερομόνης, στον οποίο η φερομόνη αφαιρείται από τα τόξα που συνθέτουν την χειρότερη λύση σε αυτήν την επανάληψη. Τα παραπάνω περιγράφονται από την εξίσωση (6.12):

$$\tau_{r,s}(t+1) = \tau_{r,s}(t)\rho' \quad , \quad \forall (r,s) \in \zeta_w \tag{6.12}$$

όπου:

ρ΄: είναι ο ρυθμός αφαίρεσης της φερομόνης εξαιτίας της αντί-φερομόνης

ζ_w: είναι η χειρότερη λύση της επανάληψης.

6.3-2 Επιθυμητή Αντί-φερομόνη (Preferential Anti- pheromone, PAP)

Ο αλγόριθμος PAP χρησιμοποιεί δύο τύπους φερομόνης. Έναν τύπο για τις καλές λύσεις και έναν για τις φτωχές λύσεις. Τα μυρμήγκια σε αυτήν την έκδοση του αλγορίθμου διαφέρουν στην προτίμηση τους για κανονική φερομόνη αντί για αντί-φερομόνη (δηλώνεται ως τ΄) σε σχέση με μια παράμετρο λ. Για κάθε μυρμήγκι k, $k=[1,m]$, η τιμή της παραμέτρου λ ισούται με $\frac{k-1}{m-1}$ [Iredi και άλλοι, 2001]. Επομένως, αντί να γίνεται βελτιστοποίηση σε δύο αντικειμενικές συναρτήσεις, ο αλγόριθμος PAP επιτρέπει σε μερικά μυρμηγκιά να εξερευνούν προφανείς φτωχές λύσεις του χώρου αναζήτησης, ενώ άλλα μυρμήγκια επικεντρώνουν το ενδιαφέρον τους στο χώρο λύσεων κοντά στην προσωρινή βέλτιστη συνολική λύση. Οι κανόνες μετάβασης που περιέχουν και την πληροφορία της αντί-φερομόνης είναι οι εξής:

$$s=\begin{cases}\arg\max_{u\in N_r^k}\left\{\left[\lambda\tau_{r,u}(t)+(1-\lambda)\tau'_{r,u}(t)\right]\left[d_{r,u}\right]^{\beta}\right\}, & \text{αν} \quad q\le q_0\\ \text{Εξίσωση (6.14)} \quad , & \text{αλλιώς}\end{cases} \tag{6.13}$$

$$P_{r,s}^{k}=\begin{cases}\dfrac{\left[\lambda\tau_{r,s}(t)+(1-\lambda)\tau'_{r,s}(t)\right]\left[d_{r,s}\right]^{\beta}}{\sum_{u\in N_r^k}\left[\lambda\tau_{r,u}(t)+(1-\lambda)\tau'_{r,u}(t)\right]\left[d_{r,u}\right]^{\beta}} & , \quad \text{αν} \quad s\in N_r^k\\ 0 & , \quad \text{αλλιώς}\end{cases} \tag{6.14}$$

Η φερομόνη και η αντί-φερομόνη ανανεώνονται εξ ίσου από τα μυρμήγκια που διασχίζουν τα τόξα κατά την διάρκεια μιας επανάληψης, καθώς η τοπική ανανέωση φερομόνης αγνοεί το κόστος των παραγόμενων λύσεων.

Η αντί-φερομόνη σε ένα επιλεγμένο τόξο ανανεώνεται ως εξής:

$$\tau'_{r,s}(t+1) = (1-\rho)\tau'_{r,s} + \rho\tau_0 \qquad (6.15)$$

Ο κανόνας συνολικής ανανέωσης της αντί-φερομόνης είναι ο εξής:

$$\tau'_{r,s}(t+1) = (1-\rho)\tau'_{r,s}(t) + \rho\Delta\tau'_{r,s}(t) \qquad (6.16)$$

με

$$\Delta\tau'_{r,s} = \begin{cases} \dfrac{Q}{L_w} & , \quad \text{αν} \quad (r,s) \in \text{διαδρομή με τη χειρότερη επανάληψη} \\ 0 & , \quad \text{αλλιώς} \end{cases}$$

όπου:

$\Delta\tau'_{r,s}(t)$: χρησιμοποιείται για να ενισχύσει την φερομόνη στα τόξα της επανάληψης με την χειρότερη λύση

L_w: είναι το μήκος της χειρότερης διαδρομής της επανάληψης, που μόλις έχει ολοκληρωθεί.

6.3-3 Εξερευνητικά μυρμήγκια

Ο αλγόριθμος των εξερευνητικών μυρμηγκιών δεν συνδέει την αντί-φερομόνη με της φτωχότερες λύσεις. Αντιθέτως, ένας μικρός αριθμός μυρμηγκιών επιλέγουν να συμπεριφερθούν διαφορετικά από άλλα μυρμήγκια τα οποία έλκονται από περιοχές με λίγη ποσότητα φερομόνης. Τα εξερευνητικά μυρμήγκια επηρεάζουν το περιβάλλον τους εναποθέτοντας φερομόνη με τον ίδιο τρόπο όπως τα κανονικά μυρμήγκια, μόνο που αντιστρέφεται η προτίμηση τους για την υπάρχουσα φερομόνη. Οι κανόνες μετάβασης είναι:

$$s=\begin{cases}\arg\max_{u\in N_r^k}\left\{\left[\tau_{max}(t)-\tau_{r,u}(t)\right]\left[d_{r,u}\right]^{\beta}\right\}, & \text{αν} \quad q\le q_0\\ \text{Εξίσωση} \quad (6.18) & , \quad \text{αλλιώς}\end{cases} \tag{6.17}$$

και

$$P_{r,s}^{k}=\begin{cases}\dfrac{\left[\tau_{max}(t)-\tau_{r,s}(t)\right]\left[d_{r,s}\right]^{\beta}}{\sum_{u\in N_r^k}\left[\tau_{max}(t)-\tau_{r,u}(t)\right]\left[d_{r,u}\right]^{\beta}} & , \quad \text{αν} \quad s\in N_r^k\\ 0 & , \quad \text{αλλιώς}\end{cases} \tag{6.18}$$

όπου:

$\tau_{max}(t)$: είναι το υψηλότερο, τρέχων, επίπεδο φερομόνης στο σύστημα.

Ο αλγόριθμος των εξερευνητικών μυρμηγκιών διαιρεί τον πληθυσμό των μυρμηγκιών σε δύο ομάδες, με μεγαλύτερη αναλογία σε κανονικά μυρμήγκια από ότι σε εξερευνητικά. Έχει βρεθεί ότι δυο εξερευνητικά μυρμήγκια παράγουν καλά αποτελέσματα όταν $m=10$.

Στον Πίνακα 6-2 δείχνονται τα αποτελέσματα της εφαρμογής των αλγορίθμων SAP, PAP και εξερευνητικών μυρμηγκιών σε ορισμένους τύπους προβλημάτων TSP [Reinelt, 1991].

Γενικά, ο SAP παράγει καλύτερες λύσεις από τον PAP. Παρόλο που ο PAP εφαρμόζεται καλύτερα από τον SAP στα προβλήματα *lin318* και *pcb442*, δεν υπάρχει στατιστικά σημαντική διαφορά. Σε προβλήματα με λιγότερες από 100 πόλεις, ο SAP παράγει καλύτερα αποτελέσματα από ότι ο αλγόριθμος των εξερευνητικών μυρμηγκιών. Όμως, σε προβλήματα με περισσότερες από 200 πόλεις, ο αλγόριθμος των εξερευνητικών μυρμηγκιών εκτελείται αποδοτικότερα. Επίσης, τα εξερευνητικά μυρμήγκια εκτελούνται καλύτερα από τον αλγόριθμο PAP σε όλα τα προβλήματα.

Πρόβλημα	Αλγόριθμος	Κόστος				CPU
		Min	Med	Max	IQR	Time
gr24	Control	1272	1278	1278	6	18
	SAP	1272	1272	1272	0	18
	PAP	1272	1272	1278	0	20
	Explorer	1272	1272	1278	5	17
eil51	Control	426	430	441	7	80
	SAP	426	428	430	1	79
	PAP	426	430	436	3	83
	Explorer	426	430	439	4	78
eil76	Control	540	545	554	8	177
	SAP	539	550	558	9	176
	PAP	539	552	562	7	182
	Explorer	539	552	561	11	172
kroA100	Control	21296	21479	22178	371	307
	SAP	21319	21552	22060	382	304
	PAP	21292	21753	22754	411	315
	Explorer	21305	21515	22318	426	298
d198	Control	15948	16116	16451	151	1181
	SAP	15988	16156	16454	337	1183
	PAP	16449	16769	17182	311	1216
	Explorer	16058	16205	16425	169	1161
lin318	Control	45514	46793	47422	1031	3018
	SAP	48375	49099	50608	1026	3050
	PAP	46793	49434	50223	954	3136
	Explorer	45031	46314	48114	908	3027
pcb442	Control	60525	62420	65014	1610	5988
	SAP	62753	64596	66332	1941	5932
	PAP	61623	64156	64753	1133	6097
	Explorer	61709	63657	66663	2219	5883

Πίνακας 6-2: Αποτελέσματα εφαρμογής των αλγορίθμων SAP, PAP, εξερευνητικών μυρμηγκιών [Montgomery και Randall, 2002].

6.4 Γενικευμένο μαθηματικό μοντέλο φερομόνης των αλγορίθμων ACO

Όταν ένα πρόβλημα συνδυαστικής βελτιστοποίησης οριστεί όπως στην παράγραφο 5-2, τότε τα τεχνητά μυρμήγκια δημιουργούν υποψήφιες λύσεις ακολουθώντας πιθανές διαδρομές, σε ένα πλήρες γράφημα $G \equiv (C, L, T)$, όπου οι κόμβοι είναι οι συνιστώσες C, το σύνολο L είναι οι

συνδέσεις (τόξα) των συνιστωσών C, ενώ T είναι το διάνυσμα ιχνών φερομόνης. Η πιθανότητα ένα μυρμήγκι, ευρισκόμενο στον κόμβο i, να επιλέξει τον κόμβο j δίνεται:

$$P\left(\mathbf{C}_{h+1}=j\middle|\mathbf{X}_h\right)=\begin{cases}\dfrac{F_{ij}\left(\tau_{ij}\right)}{\sum\limits_{(i,j)\in N_i}F_{ij}\left(\tau_{ij}\right)} & , \quad \text{αν} \quad (i,j)\in N_i^k \\ 0 & , \quad \text{αλλιώς}\end{cases} \tag{6.19}$$

όπου (i,j) ανήκει στο σύνολο N_i^k αν και μόνον εάν η ακολουθία $X_{h+1}=\{c_1,c_2,\mathbf{K},c_h,j\}$ δημιουργείται από το μυρμήγκι k που ικανοποιεί τους περιορισμούς του προβλήματος και $F_{ij}\left(\tau_{ij}\right)$ είναι μια αύξουσα, μονότονη συνάρτηση που εκφράζει τη συνάρτηση φερομόνης σε κάθε τόξο [Dorigo και Stützle, 2004]. Αν τεθεί $F_{ij}\left(\tau_{ij}\right)\equiv f_n\left(\tau_{ij}\right)$, τότε προκύπτει η παρακάτω πρόταση [Foundas και Vlachos, 2005]:

Πρόταση 6.1: Η συνάρτηση $f_n\left(\tau_{ij}\right)$ δίνεται από την ορίζουσα της μήτρας:

$$\begin{bmatrix} \alpha_n & \alpha_{n-1} & \alpha_{n-2} & \mathbf{L} & \alpha_1 & \alpha_0 \\ -1 & \tau_{ij} & 0 & \mathbf{L} & 0 & 0 \\ 0 & -1 & \tau_{ij} & \mathbf{L} & 0 & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{L} & \mathbf{M} & \mathbf{M} \\ 0 & 0 & 0 & \mathbf{L} & -1 & \tau_{ij} \end{bmatrix} \tag{6.20}$$

με $\alpha_n \geq 0, \quad n=0,1,\mathbf{K},n$

Απόδειξη: Έστω

$$f_n\left(\tau_{ij}\right)=\begin{vmatrix} \alpha_n & \alpha_{n-1} & \alpha_{n-2} & \mathbf{L} & \alpha_1 & \alpha_0 \\ -1 & \tau_{ij} & 0 & \mathbf{L} & 0 & 0 \\ 0 & -1 & \tau_{ij} & \mathbf{L} & 0 & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{L} & \mathbf{M} & \mathbf{M} \\ 0 & 0 & 0 & \mathbf{L} & -1 & \tau_{ij} \end{vmatrix} \tag{6.21}$$

Αναπτύσσοντας την ορίζουσα, κατά τα στοιχεία της $\alpha^{ης}$ στήλης προκύπτει:

$$f_n\left(\tau_{ij}\right)=\alpha_n\tau_{ij}^n+f_{n-1}\left(\tau_{ij}\right)\ \mu\varepsilon\ f_0\left(\tau_{ij}\right)=\alpha_0 \qquad (6.22)$$

και $n=1,2,\mathbf{K},n$

Προσθέτοντας αλληλοδιαδοχικά, κατά μέλη, την $f_n\left(\tau_{ij}\right)$ προκύπτει, τελικά, ότι:

$$f_n\left(\tau_{ij}\right)=\alpha_n\tau_{ij}^n+\alpha_{n-1}\tau_{ij}^{n-1}+\mathbf{L}+\alpha_1\tau_{ij}+\alpha_0 \qquad (6.23)$$

Η αναδρομική σχέση (6.23) είναι εξίσωση διαφορών πρώτου βαθμού με σταθερούς συντελεστές, της μορφής:

$$f_n\left(\tau_{ij}\right)-f_{n-1}\left(\tau_{ij}\right)=\alpha_n\tau_{ij}^n$$

και η λύση της είναι το πολυώνυμο:

$$f_n\left(\tau_{ij}\right)=\alpha_n\tau_{ij}^n+\alpha_{n-1}\tau_{ij}^{n-1}+\mathbf{L}+\alpha_1\tau_{ij}+\alpha_0,\qquad \alpha_n\geq 0$$

Πρόταση 6: Το πολυώνυμο (6.23) είναι λύση της διαφορικής εξίσωσης:

$$\left(\frac{d^{(n+1)}}{d\tau_{ij}^{(n+1)}}\right)=0 \quad (n+1)^{ης}\ \text{τάξης}$$

Απόδειξη:

Είναι:

$$\frac{d}{d\tau_{ij}}\left(\frac{d^n f_n}{d\tau_{ij}^n}\right)=0$$

οπότε

$$\frac{d^n f_n}{d\tau_{ij}^n}=c_1$$

Αν $c_1 = \alpha_n$, τότε

$$\int \frac{d}{d\tau_{ij}} \left(\frac{d^{(n-1)} f_n}{d\tau_{ij}^{(n-1)}} \right) = \int \alpha_n$$

και

$$\frac{d^{(n-1)} f_n}{d\tau_{ij}^{(n-1)}} = \alpha_n \tau_{ij} + \alpha_{n-1}$$

οπότε

$$\int \frac{d}{d\tau_{ij}} \left(\frac{d^{(n-2)} f_n}{d\tau_{ij}^{(n-2)}} \right) = \int \alpha_n \tau_{ij} + \alpha_{n-1}$$

και

$$\frac{d^{(n-2)} f_n}{d\tau_{ij}^{(n-2)}} = \alpha_n \tau_{ij}^2 + \alpha_{n-1} \tau_{ij} + \alpha_{n-2} \qquad (6.24)$$

Με αλληλοδιαδοχικές ολοκληρώσεις από την σχέση (6.24) προκύπτει, τελικά:

$$f_n(\tau_{ij}) \equiv \alpha_n \tau_{ij}^n + \alpha_{n-1} \tau_{ij}^{(n-1)} + \mathbf{L} + \alpha_1 \tau_{ij} + \alpha_0 .$$

6.5 Μαθηματικά μοντέλα διαχείρισης φερομόνης των δυο βασικών αλγορίθμων: Ant Colony System (ACS) και Max – Min Ant System (MMAS)

Σε αυτή την παράγραφο θα δημιουργηθούν μαθηματικά μοντέλα διαχείρισης φερομόνης, για τους αλγορίθμους ACS και MMAS, χρησιμοποιώντας τη θεωρία των Εξισώσεων Διαφορών [Foundas και Vlachos, 2005].

Οι εξισώσεις διαφορών εμφανίζονται σε μαθηματικά μοντέλα, όπου η μεταβλητή παίρνει ένα διακριτό σύνολο τιμών και χρησιμοποιούνται,

κυρίως, στην αριθμητική ανάλυση όπου συνεχείς συναρτήσεις προσεγγίζονται με συναρτήσεις διακριτής μεταβλητής.

6.5-1 Μαθηματικό μοντέλο τοπικής ανανέωσης φερομόνης του αλγορίθμου ACS

Κατά τη διάρκεια αναζήτησης λύσης, σ' ένα πρόβλημα συνδυαστικής βελτιστοποίησης μπορεί τα μυρμήγκια να εγκλωβιστούν σε μία και μόνο μία διαδρομή, απενεργοποιώντας τη συλλογική αναζήτηση της αποικίας. Για να αποφευχθεί ο εγκλωβισμός πρέπει κάθε μυρμήγκι που επισκέπτεται κάθε δυάδα πόλεων να αφαιρεί φερομόνη από το τόξο που συνδέει τις δυο πόλεις. Οπότε η εξερεύνηση κατευθύνεται μακριά από την καλύτερη που έχει υπολογιστεί μέχρι αυτή τη χρονική στιγμή. Σε αυτή την περίπτωση ισχύει:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \rho\tau_0 \qquad (6.25)$$

όπου:

τ_0: είναι η αρχική ποσότητα φερομόνης

ρ: είναι ο συντελεστής εξάτμισης της φερομόνης, $\rho \in (0,1)$.

Πειραματικά έχει βρεθεί ότι $\tau_0 = (nL_{nn})^{-1}$ [Dorigo και Gambardella, 1996] με n ο αριθμός των πόλεων και L_{nn} το μήκος μιας διαδρομής που υλοποιείται με τον αλγόριθμο Nearest Neighbor Heuristic.

Η εξίσωση (6.25) είναι μια εξίσωση διαφορών της μορφής:

$$y_{t+1} - (1-\rho)y_t = \rho\tau_0 \qquad (6.26)$$

όπου: $\tau_{ij}(t) \equiv y_t$

1) Η γενική λύση της ομογενούς εξίσωσης είναι:

$$y_{t+1} - (1-\rho)y_t = 0$$

με χαρακτηριστική εξίσωση:

$$\lambda - (1-\rho) = 0$$

είναι::

$$y_t = \rho(1-\rho)^t$$

2) Η μερική λύση της εξίσωσης (6.26) είναι:

$$\Psi_t = \frac{\rho\tau_0}{1+(-1+\rho)} = \tau_0$$

Η γενική λύση της εξίσωσης διαφορών είναι:

$$y_t = c(1-\rho)^t + \tau_0$$

ή

$$\tau_{ij}(t) = c(1-\rho)^t + \tau_0$$

εφόσον $\tau_{ij}(0) = 0$, τότε: $0 = c + \tau_0$ και $c = -\tau_0$.

Τελικά, η γενική λύση είναι :

$$\tau_{ij}(t) = \tau_0\left[1-(1-\rho)^t\right]$$

με $0 < \rho < 1$.

6.5-2 Μαθηματικό μοντέλο ανανέωσης φερομόνης του αλγορίθμου MMAS

Στον αλγόριθμο Max – Min Ant System (MMAS) [Stützle και Hoos, 2000] η ανανέωση της φερομόνης δίνεται από τη σχέση:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (6.27)$$

όπου:

$$\Delta\tau_{ij}(t)=\begin{cases}\dfrac{1}{L^{*}(t)}\ , & \text{αν}\quad (i,j)\in T^{*}(t)\\ 0\quad\ , & \text{αν}\quad (i,j)\notin T^{*}(t)\end{cases} \tag{6.28}$$

όπου:

$L^{*}(t)$: είναι το μήκος της συνολικής βέλτιστης διαδρομής στη χρονική στιγμή $t\left(L_{best}(t)\right)$ ή της καλύτερης λύσης κατά την επανάληψη $t\left(L_{iter}(t)\right)$.

$T^{*}(t)$: είναι η αντίστοιχη σειρά των πόλεων.

Η εξίσωση (6.27) είναι μια γραμμική μη ομογενής εξίσωση διαφορών:

$$\tau_{ij}(t+1)-(1-\rho)\tau_{ij}(t)=\Delta\tau_{ij}(t) \tag{6.29}$$

Είναι γνωστό ότι στον αλγόριθμο MMAS χρησιμοποιείται το άνω (τ_{max}) και κάτω (τ_{min}) όριο για τον περιορισμό της φερομόνης.

Πρόταση 6.3: Το άνω όριο φερομόνης στον αλγόριθμο MMAS δίνεται από τη σχέση:

$$\tau_{max}=\frac{1}{\rho L_{opt}}$$

όπου L_{opt} είναι το μήκος της πραγματικής, συνολικής βέλτιστης διαδρομής. Συνήθως αυτό δεν είναι γνωστό εκ των προτέρων γι' αυτό χρησιμοποιείται στη θέση του το $L_{best}(t)$.

Απόδειξη :

1) Η γενική λύση της ομογενούς εξίσωσης (6.29)

$$\tau_{ij}(t+1)-(1-\rho)\tau_{ij}(t)=0$$

είναι:

$$\tau_{ij}(t) = c(1-\rho)^t$$

2) Εφόσον είναι $1+(-1+\rho)=\rho>0$, η μερική λύση $T_{ij}(t)$ είναι:

$$T_{ij}(t) = \frac{\frac{1}{L_{opt}}}{\rho} = \frac{1}{\rho L_{opt}}$$

οπότε η γενική λύση είναι:

$$\tau_{ij}(t) = c(1-\rho)^t + \frac{1}{\rho L_{opt}} \qquad (6.30)$$

Για $t=0$ η εξίσωση (6.30) δίνει:

$$\tau_{ij}(0) = c + \frac{1}{\rho L_{opt}}$$

και

$$c = \tau_{ij}(0) - \frac{1}{\rho L_{opt}}$$

Η εξίσωση (6.30) παίρνει τη μορφή:

$$\tau_{ij}(t) = \left[\tau_{ij}(0) - \frac{1}{\rho L_{opt}}\right](1-\rho)^t + \frac{1}{\rho L_{opt}}$$

εφόσον $0<\rho<1$,

$$\lim_{t\to\infty}(1-\rho)^t = 0$$

και

$$\tau_{max} = \lim_{t\to\infty} \tau_{ij}(t) = \frac{1}{\rho L_{opt}}$$

Αν $\rho = 0$, η αντίστοιχη Εξίσωση Διαφορών είναι:

$$\tau_{ij}(t+1) - \tau_{ij}(t) = \frac{1}{L_{opt}}$$

Η αντίστοιχη ομογενής είναι:

$$\tau_{ij}(t+1) - \tau_{ij}(t) = 0$$

με χαρακτηριστική $\lambda - 1 = 0$, οπότε $\tau_{ij}(t) = c \equiv$ σταθ.

Η μερική λύση $T_{ij}(t)$ είναι:

$$T_{ij}(t) = \frac{\frac{1}{L_{opt}} t}{1} = \frac{1}{L_{opt}} t$$

Τελικά η γενική λύση είναι:

$$\tau_{ij}(t) = c + \frac{1}{L_{opt}} t \qquad (6.31)$$

Η σχέση (6.31) είναι ένα γραμμικό μοντέλο της φερομόνης ως προς το χρόνο.

ΚΕΦΑΛΑΙΟ 7

ΜΙΑ ΔΙΑΦΟΡΟΠΟΙΗΜΕΝΗ ΜΟΡΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ANT COLONY SYSTEM (ACS) ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (VRP)

7.1 Εισαγωγή

Ένα πρόβλημα δρομολόγησης οχημάτων (VRP) αποτελείται από ένα σύνολο πελατών που εξυπηρετούνται από ένα σύνολο οχημάτων [Thangiah και άλλοι, 2001].

Τα οχήματα που χρησιμοποιούνται συνήθως, θεωρούνται περιορισμένης χωρητικότητας φόρτωσης αγαθών με προορισμό να τα παρέχουν στους πελάτες ή να τα παραλαμβάνουν από αυτούς, αντίστοιχα. Στο κλασικό πρόβλημα δρομολόγησης οχημάτων (VRP) θεωρείται ότι όλα τα οχήματα έχουν την ίδια χωρητικότητα. Το VRP είναι εύκολο να οριστεί αλλά εξαιρετικά πολύπλοκο να επιλυθεί.

Για να κατανοηθεί η πολυπλοκότητα του προβλήματος, θεωρείστε ότι είναι αναγκαίο να επιλυθεί ένα σκέλος του VRP στο οποίο χρησιμοποιείται μια διαδρομή για την επίσκεψη ενός συνόλου πελατών.

Αν θεωρηθεί ότι το όχημα έχει απεριόριστη χωρητικότητα και χρησιμοποιείται για την εξυπηρέτηση ενός συνόλου πελατών τότε το πρόβλημα ανάγεται σε ένα πρόβλημα περιοδεύοντος πωλητή (TSP). Στο TSP ένας πωλητής ξεκινά από μία κεντρική βάση (αποθήκη), στη συνέχεια επισκέπτεται ένα σύνολο πελατών και επιστρέφει στη βάση του. Στην επίλυση τέτοιων προβλημάτων, η πολυπλοκότητα αυξάνεται εκθετικά ως προς τον αριθμό των πελατών.

Το VRP γίνεται περισσότερο πολύπλοκο από το TSP όταν κάθε διαδρομή στο VRP είναι ένα TSP. Πράγματι, αν είναι M ο αριθμός των οχημάτων σε ένα VRP, τότε για την εύρεση του καλύτερου συνόλου διαδρομών για τα M οχήματα, απαιτείται να επιλυθούν M προβλήματα περιοδεύοντος πωλητή.

Στο κεφάλαιο αυτό [Foundas και Vlachos, 2005]

- Θα οριστεί η αντικειμενική συνάρτηση ενός τυπικού VRP,
- Θα αναπτυχθούν οι παραδοχές και τα βήματα του αλγορίθμου που χρησιμοποιείται
- Θα μελετηθεί η επίδραση των μεταβολών της αρχικής ποσότητας φερομόνης στην ελαχιστοποίηση της αντικειμενικής συνάρτησης του VRP

Περιγραφή του VRP και τα μαθηματικά μοντέλα της διαφοροποιημένης μορφής του DFACS

Ένα, τυπικό, VRP ορίζεται μέσω ενός πλήρους γραφήματος $G \equiv (N, A)$, όπου:

$N = \{0,1,\mathbf{K},n\}$ είναι το σύνολο των κόμβων ενώ το 0 εκφράζει την κεντρική βάση (αποθήκη)

$A = \{(i,j) : i,j \in N, i \neq j\}$ είναι το σύνολο των τόξων που συνδέουν τους κόμους

Στους κόμβους του γραφήματος αντιστοιχούν οι πελάτες του VRP, ενώ σε κάθε τόξο $(i,j) \in N$ αντιστοιχεί ένα μη αρνητικό κόστος διάσχισης από τον κόμβο i στον κόμβο j. Θεωρείστε ότι κάθε όχημα k έχει χωρητικότητα Q_k.

Στην οικογένεια των VRP ανήκουν προβλήματα στα οποία ζητείται ο καλύτερος συνδυασμός διαδρομών ενός ορισμένου αριθμού πρακτόρων (οχημάτων), ώστε να ικανοποιούνται οι απαιτήσεις ενός, επίσης ορισμένου αριθμού κόμβων, οι οποίοι ενώνονται μεταξύ τους με διαδρομές διαφορετικού κόστους.

Στους αλγόριθμους ACO κάθε μυρμήγκι αντιστοιχεί σε ένα όχημα. Έχει μεγάλη αξία να περιγραφεί ο τρόπος με τον οποίον το μυρμήγκι επιλέγει τον επόμενο κόμβο. Αυτή η επιλογή συνδέεται με την ποσότητα της φερομόνης που εναποτίθεται σε κάθε τόξο και με το κόστος διάσχισης αντίστοιχα.

Για κάθε μυρμήγκι υπάρχουν δύο δυνατότητες δράσης. Είτε θα επιλέξει με απόλυτο τρόπο τη διαδρομή με την καλύτερη απόδοση (υψηλή ποσότητα φερομόνης, χαμηλό κόστος), είτε θα επιλέξει με βάση ένα πιθανοκρατικό κανόνα, μεταξύ των υποψηφίων διαδρομών. Στην πρώτη περίπτωση η ιδανική διαδρομή δίνεται από την παρακάτω εξίσωση [Dorigo και Gambardella, 1997]:

$$j = \begin{cases} \arg\max_{u \in S_i^k} \left\{ \left[\tau_{i,u}(t) \right] \left[n_{i,u} \right]^{\beta} \right\} & , \quad \text{αν} \quad q \le q_0 \\ \text{Εξίσωση (7.2)} & , \quad \text{αλλιώς} \end{cases} \tag{7.1}$$

$\tau_{i,u}$ είναι το ίχνος φερομόνης του τόξου (i,u),

$n_{i,u}$ είναι το αντίστροφο της απόστασης,

$n_{i,u} = \frac{1}{d_{i,u}}$ μεταξύ του κόμβου i και του κόμβου $u\left(d_{i,u}\right)$

S_i^k είναι το σύνολο των κόμβων που πρέπει να επισκεφθεί το μυρμήγκι k, τοποθετημένο στον κόμβο i,

q είναι ένας τυχαίος αριθμός ομοιόμορφα κατανεμημένος στο διάστημα $[0,1]$,

q_0 είναι μία παράμετρος $0 \le q_0 \le 1$ που καθορίζει τη σχετική σημασία της εκμετάλλευσης έναντι της εξερεύνησης.

Στη δεύτερη περίπτωση, το μυρμήγκι k επιλέγει μια τυχαία μεταβλητή j, η οποία δίνει την πιθανότητα $\left(P_{i,j}^k\right)$ με την οποία το μυρμήγκι k από τον κόμβο i θα επιλέξει τον κόμβο j η οποία ευνοεί μικρές σε κόστος διαδρομές, με υψηλή ποσότητα φερομόνης:

$$P_{i,j}^k = \begin{cases} \dfrac{\left[\tau_{i,j}(t) \right] \left[n_{i,j} \right]^{\beta}}{\sum_{u \in S_i^k} \left[\tau_{i,u}^{(t)} \right] \left[n_{i,u} \right]^{\beta}} & , \quad \text{αν} \quad j \in S_i^k \\ 0 & , \quad \text{αλλιώς} \end{cases} \tag{7.2}$$

Ποιος από τους δύο τρόπους θα προτιμηθεί από το μυρμήγκι k, δίνεται από ένα δεύτερο πιθανοκρατικό κανόνα, ο οποίος στην προκειμένη περίπτωση είναι μηδενικός για την πρώτη δυνατότητα και ίση με 1 για τη δεύτερη δυνατότητα. Σε αυτό το κεφάλαιο χρησιμοποιείται η εξίσωση (7.2) για την επιλογή του επόμενου τόξου, με $\tau_{i,j}$ να εκφράζει την ποσότητα φερομόνης που εναποτίθεται στο τόξο (i,j), ενώ $n_{i,j}$ εκφράζει το αντίστροφο του κόστους διάσχισης, $C_{i,j}$, του ίδιου τόξου, υψωμένο στη παράμετρο β. Το β ορίζει τη βαρύτητα του κόστους του κάθε τόξου στον πιθανοκρατικό κανόνα.

Στη διαφοροποιημένη μορφή του αλγορίθμου ACS για την επίλυση του VRP, η εξίσωση που εκφράζει την ανανέωση της φερομόνης σε κάθε τόξο (i,j) μετά τη διάσχισή του είναι:

$$\tau^{*}_{(i,j)} = (1-\rho)\tau_{i,j} + \tau_0 \qquad (7.3)$$

όπου:

τ_0 είναι η αρχική ποσότητα φερομόνης που αντικαθιστά το ποσό της φερομόνης που εξατμίζεται $(1-\rho)$

ρ είναι ο συντελεστής εξάτμισης φερομόνης $(0 \le \rho \le 1)$

Στο τέλος του κάθε κύκλου, υπολογίζεται η τιμή της αντικειμενικής συνάρτησης του VRP, η οποία δίνεται από την εξίσωση:

$$f_{obj} = \sum \mu_{i,j} \cdot C^{-1}_{i,j} \qquad (7.4)$$

όπου:

$\mu_{i,j}$ εκφράζει το πλήθος των διασχίσεων που έγιναν για το τόξο που ενώνει τους κόμβους i και j,

$C^{-1}_{i,j}$ είναι το κόστος διάσχισης του αντίστοιχου τόξου.

Τελικά, η τιμή της αντικειμενικής συνάρτησης προκύπτει αθροίζοντας τα επιμέρους κόστη κάθε τόξου που χρησιμοποιήθηκε, επί το πλήθος των διασχίσεών του. Σε αυτό το κεφάλαιο η παράμετρος β $(\beta = 1)$ ορίζεται από το χρήστη ενώ δεν παίρνεται υπόψη η εξάτμιση της φερομόνης.

7.2 Περιγραφή αλγορίθμου που χρησιμοποιήθηκε για την επίλυση του VRP

Αρχικά αναφέρονται οι παραδοχές που ισχύουν και στη συνέχεια παρουσιάζεται, με μορφή βημάτων, ο αλγόριθμος που χρησιμοποιήθηκε για την επίλυση του VRP.

7.2-1 Παραδοχές

- Ο κάθε πράκτορας (μυρμήγκι) δεν μεταφέρει φορτίο προς την πηγή, αλλά το αντίστροφο,
- ορισμός μεταβλητής φορτίου μυρμηγκιού, η οποία ενημερώνεται μετά από κάθε επίσκεψη σε κάθε κόμβο,
- σε κάθε επίσκεψη, θα πρέπει να ικανοποιηθεί πλήρως η απαίτηση του κόμβου σε φορτίο, ή να δοθεί όλη η ποσότητα που διαθέτει το μυρμήγκι,
- δεν υπάρχει πηγή τροφής, κάθε μυρμήγκι μπορεί να μετακινηθεί σε οποιοδήποτε κόμβο, εφόσον σ' αυτό υπάρχει ζήτηση,
- κόμβοι στους οποίους η ζήτηση έχει καλυφθεί, αγνοούνται από τα μυρμήγκια (αλλά μπορούν να είναι ενδιάμεσοι σταθμοί, αν αυτό ευνοεί τη μετάβαση σε έναν κόμβο στον οποίο υπάρχει ζήτηση),
- κάθε μυρμήγκι που μένει χωρίς φορτίο, επιστρέφει στην αφετηρία (το κόστος επιστροφής δεν συνυπολογίζεται, με βάση τις απαιτήσεις του προβλήματος),
- η απαγόρευση της δεύτερης επίσκεψης σε έναν κόμβο ισχύει μόνο μέχρι την επιστροφή στην πηγή για ανεφοδιασμό,
- όταν καλυφθεί η ζήτηση σε όλους τους κόμβους, τα μυρμήγκια επιστρέφουν αυτόματα στην αφετηρία και γίνεται η ανανέωση των ιχνών φερομόνης σε κάθε διαδρομή, ανάλογα με το πόσο αυτή χρησιμοποιήθηκε.

7.2-2 Αλγόριθμος

1. Εισαγωγή του γραφήματος του προβλήματος, μέσω των παρακάτω δεδομένων:
   α. Πλήθος κόμβων

   β. Εισαγωγή τόξων με τη μορφή τετράδας αριθμών, στην οποία ο πρώτος αριθμός αναπαριστά τον αύξοντα αριθμό του κόμβου εκκίνησης, ο δεύτερος αυτόν του κόμβου τερματισμού, ο τρίτος το κόστος της διαδρομής και το τέταρτος την απαίτηση του τερματικού κόμβου σε φορτίο.

2. Αρχικοποίηση με:

   α. Τοποθέτηση μίας αρχικής ποσότητας φερομόνης σε όλους τους κόμβους

   β. Ορισμός του φορτίου που μπορεί να μεταφέρει το κάθε μυρμήγκι

   γ. Ορισμός πλήθους μυρμηγκιών, τέτοιου ώστε μόλις να καλύπτεται το άθροισμα των απαιτήσεων των κόμβων σε φορτίο

3. Επανέλαβε μέχρι να βρεθεί ικανοποιητική λύση

   3.1 Επανέλαβε για κάθε μυρμήγκι i

      3.1.1 Αν το φορτίο του μυρμηγκιού είναι μηδενικό, τότε επιστρέφει στην αφετηρία

      3.1.2 Αλλιώς επανέλαβε για κάθε κόμβο j

         3.1.2.1 Από τον κόμβο $j-1$ προς τον κόμβο j και εφόσον ισχύουν τα:

            Υπάρχει το τόξο $(j-1, j)$

            Υπάρχει ζήτηση φορτίου από τον κόμβο j

            Το μυρμήγκι i δεν έχει επισκεφθεί ξανά τον κόμβο j από τη στιγμή της τελευταίας εκκίνησης (αν ισχύει το II ισχύει πάντα και το III) τότε υπολόγισε την πιθανότητα επιλογής του κόμβου j από το μυρμήγκι

3.1.3 Τέλος 3.1.2

3.1.4 Το μυρμήγκι i επιλέγει τον κόμβο προς επίσκεψη, με βάση τις επιμέρους πιθανότητες που υπολογίστηκαν

3.2 Τέλος 3.1

3.3 Υπολογισμός τιμής αντικειμενικής συνάρτησης

3.4 Επαναϋπολογισμός της ποσότητας φερομόνης σε κάθε τόξο.

4. Τέλος

7.3 Αποτελέσματα εφαρμογής

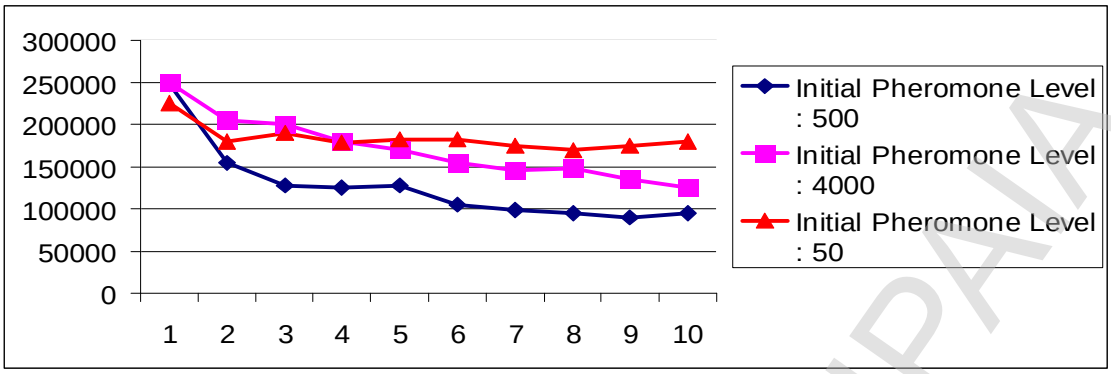
Χρησιμοποιώντας το πρόγραμμα του Παραρτήματος Ι δείχνεται πώς οι έννοιες που περιγράφτηκαν παραπάνω μπορούν να υλοποιηθούν πρακτικά μέσω του προτεινόμενου αλγορίθμου.

Ο αλγόριθμος υλοποιήθηκε για ένα γράφημα οκτώ (8) κόμβων με αντίστοιχες απαιτήσεις και για αρχικές ποσότητες φερομόνης όπως δείχνεται στον πίνακα 7.1

Κόμβος	Απαίτηση
1	0
2	500
3	900
4	200
5	90
6	800
7	1000
8	50

$$\tau_0 = 500, 4000, 50$$
$$Q_k = 50$$

Στο Σχήμα 1 δείχνεται το αποτέλεσμα της μεταβολής της αρχικής ποσότητας φερομόνης στη βελτιστοποίηση της αντικειμενικής συνάρτησης στο χρόνο. Στην παρούσα υλοποίηση ο χρόνος μετριέται σε κύκλους.



Σχήμα 7.1: Επίδραση μεταβολής της αρχικής ποσότητας φερομόνης στη βελτιστοποίηση της αντικειμενικής συνάρτησης του VRP

Από το σχήμα 7.1 προκύπτει ότι για $\tau_0 = 4000$ παρατηρείται μια, σχετικά, σταθερή μείωση της τιμής της αντικειμενικής συνάρτησης, φτάνοντας, όμως, στο δέκατο κύκλο δεν υπάρχει κάποια σταθεροποίηση ή κάποιο όριο στο οποίο να τείνει σαφώς η τιμή αυτή.

Για $\tau_0 = 500$ προκύπτουν τα καλύτερα αποτελέσματα της βελτιστοποίησης. Στο τέλος του πρώτου κύκλου, η τιμή της αντικειμενικής συνάρτησης περιλαμβάνεται μεταξύ των $\tau_0 = 4000$ και $\tau_0 = 50$. Στη συνέχεια παρατηρείται μια πολύ καλή βελτιστοποίηση της αρχικής τιμής με την αντικειμενική συνάρτηση να μειώνεται με εκθετικό ρυθμό και να φτάνει σε μια σχεδόν σταθερή κατάσταση στο τέλος του δέκατου κύκλου.

Για το $\tau_0 = 50$, παρατηρείται πως το αποτέλεσμα της αντικειμενικής συνάρτησης στο τέλος του πρώτου κύκλου είναι χαμηλότερο απ' ότι στις άλλες δύο περιπτώσεις, αλλά παρατηρείται πολύ μικρή ή καθόλου βελτίωση στο τέλος των υπολοίπων κύκλων.

7.4 Γενικές παρατηρήσεις

Α) Μελετώντας τα παραπάνω αποτελέσματα εφαρμογής, προκύπτουν παρατηρήσεις, που ισχύουν γενικότερα στους αλγορίθμους αποικίας μυρμηγκιών. Είναι προφανές ότι η ποιότητα των αποτελεσμάτων είναι άμεσα συναρτώμενη των αρχικών τιμών της φερομόνης. Παρατηρούνται σημαντικές διακυμάνσεις των αποτελεσμάτων τροποποιώντας ελάχιστα τις αρχικές ποσότητες φερομόνης.

Σημαντικός παράγοντας σε οποιαδήποτε υλοποίηση του αλγορίθμου αποικίας μυρμηγκιών είναι ο σωστός ορισμός της αρχικής ποσότητας

φερομόνης, που θα κρίνει το κατά πόσο θα ευνοούνται τα καλύτερα μονοπάτια έναντι των υπολοίπων. Η αρχική ποσότητα φερομόνης ορίζει την ευελιξία του συστήματος στην εύρεση νέων, κάθε φορά, λύσεων.

Παρατηρείται, επίσης ότι μεγαλύτερος αριθμός κύκλων προσομοίωσης οδηγεί σε μεγαλύτερη προσέγγιση της ιδανικής λύσης. Όταν, μάλιστα, η λύση αυτή έχει προσεγγιστεί με μεγάλη ακρίβεια, η τιμή της αντικειμενικής συνάρτησης παύει να έχει πτωτική πορεία και ταλαντεύεται μεταξύ ενός σημείου. Στο σημείο αυτό, είναι πολύ δύσκολο (αν και όχι απίθανο) να βρεθεί μία σημαντική καλύτερη λύση και λαμβάνεται η χαμηλότερη τιμή που έχει βρεθεί μέχρι τη στιγμή του τερματισμού ως η βέλτιστη.

Β) Οι τρεις παράμετροι που πρέπει να οριστούν πριν από την εκκίνηση της λειτουργίας του αλγορίθμου DFACS είναι:

- το μέγιστο φορτίο που μπορεί να μεταφέρει ο κάθε πράκτορας (μυρμήγκι)
- το πλήθος των κύκλων «τρεξίματος» του DFACS και
- η αρχική ποσότητα της φερομόνης.

Η πρώτη παράμετρος επηρεάζει το πλήθος των διαδρομών που θα απαιτηθούν ώστε να ικανοποιηθεί πλήρως η απαίτηση κάθε κόμβου σε φορτίο. Όσο μεγαλύτερη είναι η ποσότητα που κάθε μυρμήγκι μπορεί να μεταφέρει, τόσα λιγότερα ταξίδια θα χρειαστεί να κάνει μέχρι το τέλος του κάθε κύκλου.

Το πλήθος των κύκλων ορίζεται σαν συνάρτηση της ποιότητας των αποτελεσμάτων που θα προκύψουν από τη λειτουργία του αλγορίθμου DFACS.

Στο τέλος κάθε κύκλου, τα δεδομένα του γραφήματος (απαντήσεις κόμβων και πλήθη διαδρομών ανά τόξο) επανέρχονται στις αρχικές του τιμές. Αυτό που παραμένει, είναι η ποσότητα της φερομόνης που έχει τοποθετηθεί σε κάθε τόξο κατά τη διάρκεια υλοποίησης του αλγορίθμου DFACS. Αυτό σημαίνει ότι, στον επόμενο κύκλο, θα διατηρηθεί η προτίμηση των πρακτόρων προς τα τόξα με μικρό κόστος, όπως αυτή διαμορφώθηκε πριν, ενώ θα συνεχίσει να βελτιώνεται. Όσο περισσότεροι κύκλοι ορίζονται, τόσο καλύτερα προσεγγίζεται η βέλτιστη λύση του προβλήματος, με κόστος, όμως, τον περισσότερο χρόνο που θα χρειαστεί για την ολοκλήρωση της υλοποίησης του αλγορίθμου DFACS.

ΚΕΦΑΛΑΙΟ 8

ΔΙΑΦΟΡΟΠΟΙΗΜΕΝΗ ΜΟΡΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ANT COLONY SYSTEM (DFACS): ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΔΙΚΤΥΟΥ ΠΛΟΙΩΝ

8.1 Εισαγωγή

Η Ελλάδα διαθέτει 14854 χιλιόμετρα ακτής, 3.500 νησιά (200 εκ των οποίων είναι ακατοίκητα) και 750 λιμάνια και αγκυροβόλια. Ο χώρος που καταλαμβάνουν αυτά τα νησιά αποτελεί το 19% της έκτασης της Ελλάδας και κατοικούνται από το 14% του πληθυσμού. Η μεταφορά των αγαθών γίνεται με φορτηγά, τα οποία μεταφέρουν τα αγαθά από τον τόπο προέλευσης στον τόπο προορισμού μέσω επιβατικών πλοίων και ένα στόλο από μικρά φορτηγά πλοία τα οποία μεταφέρουν κυρίως εμπορεύματα, τα οποία αποτελούν ένα μικρό ποσοστό αυτής της δραστηριότητας. Η ακτοπλοϊκή μεταφορά στην Ελλάδα αποτελεί το πιο σημαντικό μέσο μεταφοράς αγαθών και το κυριότερο μέσο μεταφοράς επιβατών.

Οι καλύτεροι τρόποι μεταφοράς εμπορευμάτων, όπως μελετήθηκε και σε επίπεδο Ευρωπαϊκής Ένωσης, στο εσωτερικό των χωρών και ειδικά στην περίπτωση της Ελλάδας με τη μεγάλη γεωγραφική διασπορά των νησιών και την ορεινή της μορφολογία, αποτελούν τα μικρά containers.

Σε αυτή την εργασία θεωρείται ένα πιλοτικό δίκτυο 13 λιμανιών (συμπεριλαμβανομένου και του κέντρου ανεφοδιασμού [Sambracos, 2000] του Αιγαίου και 34 συνδέσεις μεταξύ των νησιών. Υποθέτουμε ότι τα μικρά containers είναι το μόνο φορτίο που μεταφέρεται. Ο στόχος μας είναι – κάτω από γνωστούς περιορισμούς – να ελαχιστοποιηθεί το συνολικό κόστος καυσίμων και έξοδα λιμανιών, τα οποία εκφράζονται στο πρόβλημα μέσα από την αντικειμενική συνάρτηση. Για την επίτευξη αυτού του στόχου χρησιμοποιείται μία διαφοροποιημένη μορφή του Αλγορίθμου Ant Colony System (DFACS) για την επίλυση προβλημάτων δρομολόγησης οχημάτων (VRP) [Foundas και Vlachos, 2005].

8.2 Περιγραφή του προβλήματος

θεωρείστε ένα μικρό υποσύνολο των νήσων του Αιγαίου, τα οποία σχηματίζουν τους κόμβους του γραφήματος του σχήματος 8.1 [Alexandris και άλλοι, 2005]. Τα τόξα που ενώνουν τους κόμβους αντιπροσωπεύουν τις υπάρχουσες (ή τις δυνατές) ακτοπλοϊκές γραμμές μεταξύ των νήσων. Κάθε νησί, ή κόμβος του γραφήματος, έχει μια συγκεκριμένη απαίτηση από containers, τα οποία παρέχονται από το διαθέσιμο ναυτιλιακό στόλο, ο οποίος τα μεταφέρει από το λιμάνι του Πειραιά, το οποίο ορίζεται ως η αφετηρία κάθε δυνατής διαδρομής στο δίκτυο. Για κάθε μια από αυτές τις ακτοπλοϊκές γραμμές υπάρχει ένα κόστος διάσχισης, το οποίο είναι ανάλογο της απόστασης των νησιών που τα ενώνει.

Στο κάθε ναυτικό μίλι αυτής της απόστασης, το κάθε πλοίο καταναλώνει μια ορισμένη ποσότητα καυσίμου, ενώ στο συνολικό κόστος θα πρέπει να προστεθεί και το κόστος επίσκεψης στο λιμάνι, το οποίο ποικίλει από νησί σε νησί.

Για χάρη απλότητας και χρηστικότητας, το κόστος επιστροφής κάθε πλοίου στο λιμάνι του Πειραιά, μετά την ολοκλήρωση της διαδρομής του, δεν υπολογίζεται.

Παρακάτω ορίζονται οι παραδοχές και οι περιορισμοί του προβλήματος:

n_{ij} ο αριθμός των πλοίων που ταξιδεύουν από τον κόμβο i στον κόμβο j,

L_{ij} το μήκος του τόξου (i, j) σε μίλια,

c_f το κόστος των καυσίμων που καταναλώνεται από μίλι (€/μίλι),

c_{pi} το κόστος χρήσης για το λιμάνι i (σε €),

Q η χωρητικότητα των πλοίων (αριθμός των containers) – θεωρείται σταθερό για όλους τους τύπους των πλοίων,

w_{ij} ο αριθμός των containers που μεταφέρθηκαν από τον κόμβο i στον κόμβος j,

D_i η απαίτηση στο λιμάνι i (αριθμός των containers),

S_i η παροχή στο λιμάνι i (αριθμός των containers).

Η αντικειμενική συνάρτηση του προβλήματος είναι:

$$\min \sum_{\substack{\text{all arcs} \\ ij}} \left[n_{ij} L_{ij} c_f + n_{ij} c_{pi} \right] \tag{8.1}$$

Παίρνονται υπόψη οι παρακάτω περιορισμοί:

$$\sum_{j=1}^{J} W_{ij} - \sum_{k=1}^{K} W_{ik} + S_i - D_i = 0, \tag{8.2}$$

ισοζύγιο των εισερχομένων και εξερχομένων containers στον κόμβο i (η απαίτηση πρέπει να ικανοποιείται σε όλους τους κόμβους)

$$n_{ij} Q - W_{if} \geq 0, \quad \forall \text{ τόξο ij}, \tag{8.3}$$

ορίζοντας τον αριθμό των πλοίων (n_{ij}) επί της χωρητικότητάς τους (Q) μεγαλύτερο ή ίσο με τον αριθμό των containers που μεταφέρθηκαν (W_{ij}), δίνεται ένα ανώτερο όριο για W_{ij}

$$W_{if} - (n_{ij} - 1) Q \geq 0, \quad \forall \text{ τόξο ij} \tag{8.4}$$

δίνει τα χαμηλότερα όρια για το W_{ij}

$$W_{if} \geq n_{ij}, \quad \forall \text{ τόξο ij} \tag{8.5}$$

διασφαλίζει ότι το W_{ij} παίρνει θετική τιμή όπου το n_{ij} γίνεται μεγαλύτερο του μηδενός

$$n_{ij} \leq n_{max}, \quad \forall \text{ τόξο ij} \tag{8.6}$$

ορίζει το ανώτερο όριο του n_{ij} (αλλιώς η αξία του n_{ij} είναι χωρίς όρια και το πρόβλημα δεν μπορεί να λυθεί)

$$n_{ij} \geq 0, \quad \forall \text{ τόξο ij} \tag{8.7}$$

μη αρνητικότητα περιορισμών

$W_{if} \geq 0, \quad \forall$ τόξο ij (8.8)

μη αρνητικότητα περιορισμών

n_{ij} ακέραιος (8.9)

ακεραιότητα

8.3 Μαθηματικά μοντέλα

Η ελαχιστοποίηση της αντικειμενικής συνάρτησης που εκφράζεται από την εξίσωση 8.1) γίνεται μέσω της διαφοροποιημένης μορφής του αλγορίθμου ACS (DFACS) για το πρόβλημα δρομολόγησης οχημάτων [Foundas και Vlachos, 2005].

Ισχύουν τα μαθηματικά μοντέλα που εκφράζονται από τις εξισώσεις:

$$j = \begin{cases} \arg\max_{u \in S_i^k} \left\{ \left[\tau_{i,u}(t) \right] \left[n_{i,u} \right]^{\beta} \right\} & , \quad \alpha\nu \quad q \leq q_0 \\ \text{Εξίσωση (8.11)} & , \quad \text{αλλιώς} \end{cases} \tag{8.10}$$

$$P_{i,j}^k = \begin{cases} \dfrac{\left[\tau_{i,j}(t) \right] \left[n_{i,j} \right]^{\beta}}{\sum_{u \in S_i^k} \left[\tau_{i,u}^{(t)} \right] \left[n_{i,u} \right]^{\beta}} & , \quad \alpha\nu \quad j \in S_i^k \\ 0 & , \quad \text{αλλιώς} \end{cases} \tag{8.11}$$

$$\tau_{(i,j)} = (1-\rho)\tau_{i,j} + \tau_0 \tag{8.12}$$

Στην εξίσωση

$$f_{obj} = \sum \mu_{i,j} \cdot C_{i,j}^{-1} \tag{8.13}$$

το $\mu_{i,j}$ αντιστοιχεί στο πλήθος των διασχίσεων για το τόξο που συνδέει τους κόμβους i και j, ενώ το $C_{i,j}^{-1}$ είναι το κόστος διάσχισης του ίδιου του τόξου. Είναι:

$$C_{i,j}^{-1} = L_{ij} C_F + C_{pj} \text{ με } i \neq j \tag{8.14}$$

Στην παρούσα υλοποίηση, η παράμετρος β πήρε την τιμή 1,8, καθώς με βάση εμπειρικών παρατηρήσεων από την εκτέλεση του προγράμματος, η τιμή αυτή έδωσε μια ικανοποιητική καμπύλη προόδου όσον αφορά τη μείωση της αντικειμενικής συνάρτησης.

8.4 Περιγραφή του αλγορίθμου (DFACS)

8.4-1 Παραδοχές

- Κάθε μυρμήγκι (πλοίο) ξεκινάει από έναν ορισμένο κόμβο (αφετηρία – λιμάνι του Πειραιά), φορτωμένο με το μέγιστο δυνατό φορτίο,

- σε κάθε επίσκεψη, θα πρέπει να ικανοποιηθεί πλήρως η απαίτηση του νησιού σε μικρά containers ή να δοθεί όλη η ποσότητα που διαθέτει το πλοίο,

- λιμάνια στα οποία η ζήτηση έχει καλυφθεί, αγνοούνται από τα πλοία (αλλά μπορούν να είναι ενδιάμεσοι σταθμοί, αν αυτό ευνοεί τη μετάβαση σε ένα νησί στο οποίο υπάρχει ζήτηση),

- κάθε πλοίο χωρίς φορτίο, επιστρέφει στην αφετηρία (το κόστος της επιστροφής δεν συνυπολογίζεται, με βάση τις απαιτήσεις του προβλήματος).

8.4-2 Ο αλγόριθμος

1. Επανέλαβε μέχρι να βρεθεί ικανοποιητική λύση.

   αν το φορτίο του πλοίου είναι μηδενικό, τότε επιστρέφει στην αφετηρία,

   αλλιώς επανέλαβε για κάθε νησί j

   1.2.1.1 Από κάθε νησί $j-1$ προς το νησί j, και εφόσον ισχύουν τα:

   i. υπάρχει η ακτοπλοϊκή γραμμή $(j-1, j)$

   ii. υπάρχει ζήτηση φορτίου από το νησί j

iii. το πλοίο i δεν έχει επισκεφθεί ξανά το νησί j (αν ισχύει το ii ισχύει και το iii)

τότε

υπολογίστε την πιθανότητα επιλογής του νησιού j από το πλοίο.

τέλος 1.1.2,

το πλοίο i επιλέγει τον κόμβο προς επίσκεψη, με βάση τον πιθανοκρατικό κανόνα που υπολογίστηκε,

τέλος 1.1

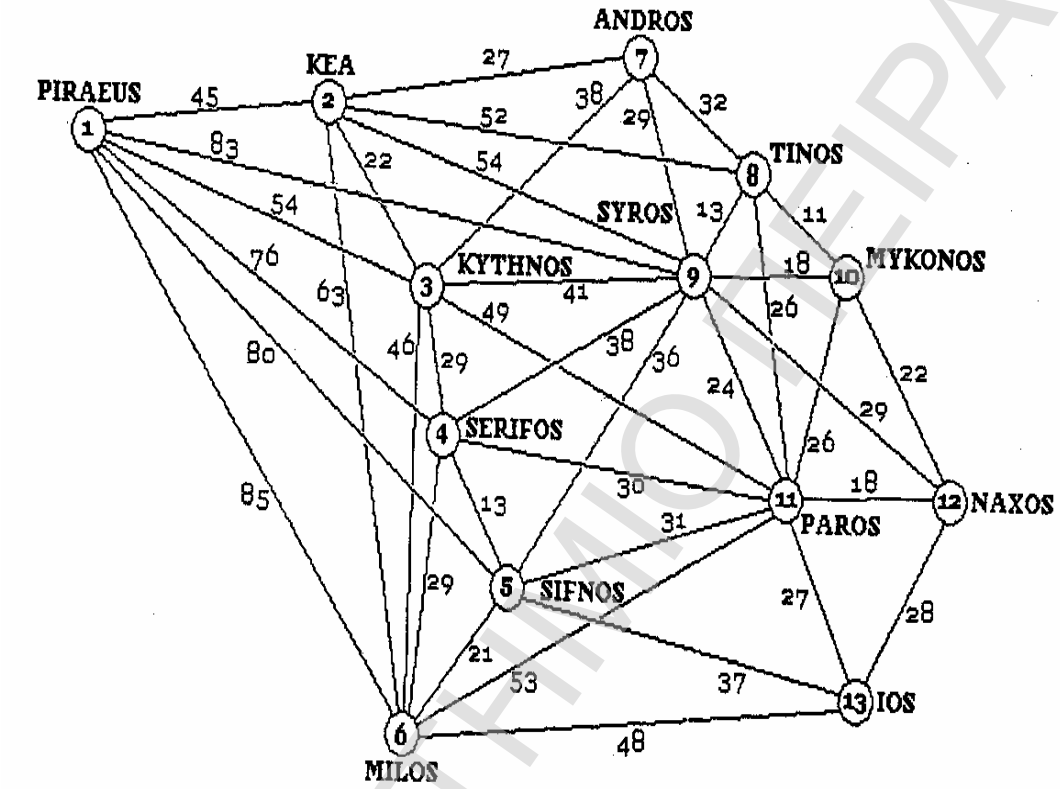
υπολογισμός της τιμής της αντικειμενικής συνάρτησης,

επαναϋπολογισμός της ποσότητας της φερομόνης σε κάθε τόξο,

2. τέλος του αλγορίθμου.

8.5 Εφαρμογή του αλγόριθμου (DFACS)

Η εφαρμογή αφορά τη βελτιστοποίηση λειτουργίας ενός δικτύου πλοίων, τα οποία μεταφέρουν μικρά containers. Το φορτίο των πλοίων μεταφέρεται σε ένα σύμπλεγμα 13 νησιών του Αιγαίου, τα οποία αποτελούν τους κόμβους απαιτήσεων του Σχ. 8.1. Οι κόμβοι διασυνδέονται με 39 τόξα ή ναυτιλιακές γραμμές. Οι αποστάσεις των νησιών του Σχ. 8.1 προέρχονται από τους ναυτιλιακούς χάρτες του Υπουργείου Εμπορικής Ναυτιλίας (M.C.M., personal communication, 2002].



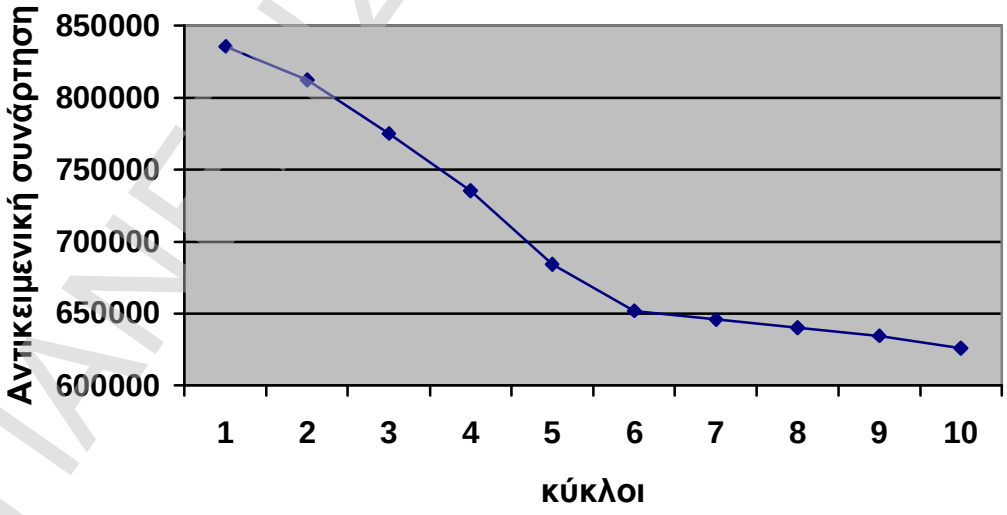
Σχήμα 8.1 Γράφημα νησιών του Αιγαίου

Υποτίθεται ότι όλος ο εφοδιασμός παρέχεται από το λιμάνι του Πειραιά. Οι απαιτήσεις τίθενται από τα νησιά που περιλαμβάνονται στο δίκτυο, με ποσότητες ανάλογες προς τους πραγματικούς τόνους που φορτώνονται και ξεφορτώνονται ετησίως βασιζόμενοι στα στοιχεία του Υπουργείου Εμπορικής Ναυτιλίας. Οι ποσότητες αυτές παρουσιάζονται στον πίνακα 8.1. Όλα τα πλοία υποτίθεται ότι έχουν μία χωρητικότητας 300 μικρών containers των 1500 τόνων, όπως δείχνεται στον πίνακα 8.2

Πίνακας 8.1 Port demand and supply

		PORT DEMAND small containers		**PORT SUPPLY** small containers
1 Πειραιάς	D_1	0	S_1	331368
2 Κέα	D_2	8002	S_2	0
3 Κύθνος	D_3	1684	S_3	0
4 Σέριφος	D_4	2992	S_4	0
5 Σίφνος	D_5	6805	S_5	0
6 Μήλος	D_6	15875	S_6	0
7 Άνδρος	D_7	1847	S_7	0
8 Τήνος	D_8	26495	S_8	0
9 Σύρος	D_9	29477	S_9	0
10 Μύκονος	D_{10}	48805	S_{10}	0
11Πάρος	D_{11}	23116	S_{11}	0
12 Νάξος	D_{12}	20251	S_{12}	0
13 Ίος	D_{13}	5066	S_{13}	0
		190415		**331368**

Η ελαχιστοποίηση της αντικειμενικής συνάρτησης δείχνεται στο σχ. 8.2



Σχήμα 8.2 Ελαχιστοποίηση της αντικειμενικής συνάρτησης στο χρόνο (κύκλοι)

Πίνακας 82. Αριθμός διαδρομών

	n πλοία	**W** Containers		**n** πλοία	**W** Containers
1-2	38	11410	2-1	0	0
1-3	78	23391	3-1	0	0
1-4	10	2992	4-1	0	0
1-5	53	15887	5-1	0	0
1-6	40	12004	6-1	0	0
1-9	417	125087	9-1	0	0
2-3	5	1500	3-2	0	0
3-4	0	0	4-3	0	0
4-5	0	0	5-4	0	0
5-6	13	3900	6-5	0	0
2-6	0	0	6-2	0	0
4-6	0	0	6-4	0	0
2-7	7	2086	7-2	0	0
2-8	0	0	8-2	0	0
2-9	0	0	9-2	0	0
7-8	0	0	8-7	0	0
7-9	0	0	9-7	0	0
3-7	0	0	7-3	0	0
3-9	0	0	9-3	0	0
8-9	0	0	9-8	89	26675
8-10	0	0	10-8	0	0
8-11	0	0	11-8	0	0
9-10	163	48895	10-9	0	0
9-12	68	20401	12-9	0	0
4-9	0	0	9-4	0	0
4-11	0	0	11-4	0	0
5-9	0	0	9-5	0	0
3-11	77	23116	11-3	0	0
10-12	0	0	12-10	0	0
10-11	0	0	11-10	0	0
11-12	0	0	12-11	0	0
9-11	0	0	11-9	0	0
5-11	0	0	11-5	0	0
6-13	0	0	13-6	0	0
12-13	0	0	13-12	0	0
5-13	17	5096	13-5	0	0
11-13	0	0	13-11	0	0
6-11	0	0	11-6	0	0
3-6	0	0	6-3	0	0

Όταν ο αλγόριθμος ξεκινά βρίσκοντας μια πρώτη λύση στο πρόβλημα, στη συνέχεια ενημερώνει κάθε τόξο με τη νέα ποσότητα φερομόνης, ανάλογης με το πλήθος των διασχίσεων του και με το κόστος του, ενώ η διαδικασία επαναλαμβάνεται με τα νέα πλέον δεδομένα που έχουν προκύψει από την αμέσως, προηγούμενη λειτουργία του αλγορίθμου.

Όταν συμπληρωθούν δέκα (10) κύκλοι επαναλήψεων της παραπάνω διαδικασίας, οι διαδοχικές τιμές που απέκτησε η αντικειμενική συνάρτηση του προβλήματος δείχνεται στο σχ. 8.2

ΚΕΦΑΛΑΙΟ 9

ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ ΣΤΑ ΚΕΝΤΡΑ ΕΛΕΓΧΟΥ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

9.1 Εισαγωγή

Το πρόβλημα της οικονομικής κατανομής ηλεκτρικού φορτίου (ELD) είναι από τα σπουδαιότερα που σχετίζονται με τη λειτουργία των κέντρων ελέγχου ενέργειας. Πολλά προβλήματα βελτιστοποίησης των συστημάτων ηλεκτρικής ισχύος, συμπεριλαμβανομένου και του ELD, έχουν σύνθετη δομή με μη γραμμικά χαρακτηριστικά και περιλαμβάνουν περιορισμούς ισοζυγίου ισχύος και τεχνικών ορίων των γεννητριών αντίστοιχα.

Η μεγάλη πίεση στην αγορά της παροχής ηλεκτρικής ενέργειας προσφέρει μεγάλα κίνητρα για την ανάπτυξη, καινούργιων, βέλτιστων αλγορίθμων που αποσκοπούν στο σχεδιασμό στρατηγικών για καλύτερη διαχείριση της ηλεκτρικής ενέργειας. Προς αυτή την κατεύθυνση έχουν ερευνηθεί και υιοθετηθεί διάφορες αλγοριθμικές προσεγγίσεις όπως: η μέθοδος Lagrange [Guan και άλλοι, 1995], ο γραμμικός και ο δυναμικός προγραμματισμός [Chebbo και άλλοι, 1995] και άλλοι μαθηματικοί μέθοδοι.

Τα τελευταία χρόνια αναπτύχθηκαν, επιτυχώς ευρεστικές τεχνικές οι οποίες υλοποιούνται μέσω κατάλληλων αλγορίθμων, όπως: γενετικοί αλγόριθμοι [Contaxis και άλλοι, 2001], αλγόριθμοι προσομοιωμένης ανάπτυξης [Wong και άλλοι, 1993], εξελικτικές στρατηγικές [Wong και άλλοι, 1998] και αλγόριθμοι Tabu Search [Mantawy και άλλοι, 1998] οι οποίοι συνεισφέρουν κατά σημαντικό τρόπο στη λύση των προβλημάτων ELD.

Πολύ πρόσφατα, η έρευνα κατευθύνθηκε προς τους μετα-ευρεστικούς αλγορίθμους και ειδικά σε αυτούς που εμπνέονται από φυσικά, βιολογικά ή κοινωνικά φαινόμενα [Corne και άλλοι, 1999]. Οι αλγόριθμοι αυτοί δεν εγκλωβίζονται σε τοπικές βέλτιστες λύσεις ενώ έχουν την ιδιότητα να εντοπίζουν την περιοχή της συνολικής βέλτιστης λύσης μετά από ένα σημαντικό αριθμό εκτίμησης της αντικειμενικής συνάρτησης.

Σε αυτή την κατηγορία των αλγορίθμων ανήκει ο αλγόριθμος Max –

Min Ant System [Stützle και άλλοι, 2001] της οικογένειας αποικίας μυρμηγκιών, ο οποίος προσαρμόζεται κατάλληλα για τη λύση του προβλήματος ELD.

9.2 Ορισμός του προβλήματος ELD

Με τον όρο οικονομική κατανομή ηλεκτρικού φορτίου εννοούμε την ελαχιστοποίηση του κόστους λειτουργίας των μονάδων παραγωγής (9.1) κάτω από τον περιορισμό ισοζυγίου ισχύος (9.2) και επίσης, κάτω από τον περιορισμό των τεχνικών ορίων των γεννητριών (9.3) [Contaxis και άλλοι, 2001]:

$$\text{Min } F_T(P) = \sum_{i=1}^{n} F_i(P_i) \tag{9.1}$$

$$\sum_{i=1}^{n} P_i = P_D \tag{9.2}$$

$$P_{i,min} \le P_i \le P_{i,max} \tag{9.3}$$

όπου:

n: είναι ο αριθμός (on-line) θερμικών μονάδων,

P_i: είναι η ισχύς εξόδου της i-οστής μονάδας σε MW,

P: είναι ένα διάνυσμα που περιέχει όλα τα P_i,

$F_i(P_i)$: είναι το κόστος παραγωγής P_i, MW, σε χιλιάδες δραχμές ανά ώρα (KGrd/h),

$P_{i,min}$ και $P_{i,max}$: είναι τα όρια ισχύος της i-οστής μονάδας σε MW,

P_D: είναι η συνολική ζήτηση φορτίου σε MW

Στη μελέτη που ακολουθεί μία από τις μονάδες παραγωγής ορίζεται από την αρχή, ως μονάδα αναφοράς. Ο υπολογισμός του κόστους που

αντιστοιχεί σε κάθε μονάδα για την παραγωγή ηλεκτρικής ισχύος δίνεται από την εξίσωση (9.4):

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \qquad (9.4)$$

όπου a_i, b_i, c_i είναι οι συντελεστές κόστους της i-οστής γεννήτριας.

9.3 Περιγραφή του αλγορίθμου που χρησιμοποιήθηκε για τη λύση του προβλήματος

Το πρόβλημα ELD είναι συνεχές. Ο αλγόριθμος Max – Min που χρησιμοποιείται εφαρμόζεται σε διακριτά προβλήματα. Οπότε για κάθε γεννήτρια διαμερίζεται το διάστημα των ορίων ισχύων της σε ένα σύνολο διακριτών τιμών. Η διαμέριση μπορεί να γίνει με διάφορους τρόπους. Στην παρούσα μελέτη διαμερίζονται όλα τα διαστήματα ισχύων σε ίσο αριθμό υποδιαστημάτων.

Τελικά, για κάθε γεννήτρια (εκτός από τη γεννήτρια αναφοράς που ορίζεται από την αρχή) προκύπτει ένα διακριτό πεπερασμένο σύνολο διαστημάτων ισχύων ανάλογα με τον τρόπο διαμέρισης.

Στην παρούσα υλοποίηση όλες οι γεννήτριες πρέπει να λειτουργούν εντός των ορίων τους ισχύων με εξαίρεση τη γεννήτρια αναφοράς που μπορεί να είναι εκτός των ορίων ισχύων παίρνοντας κάποια ποινή.

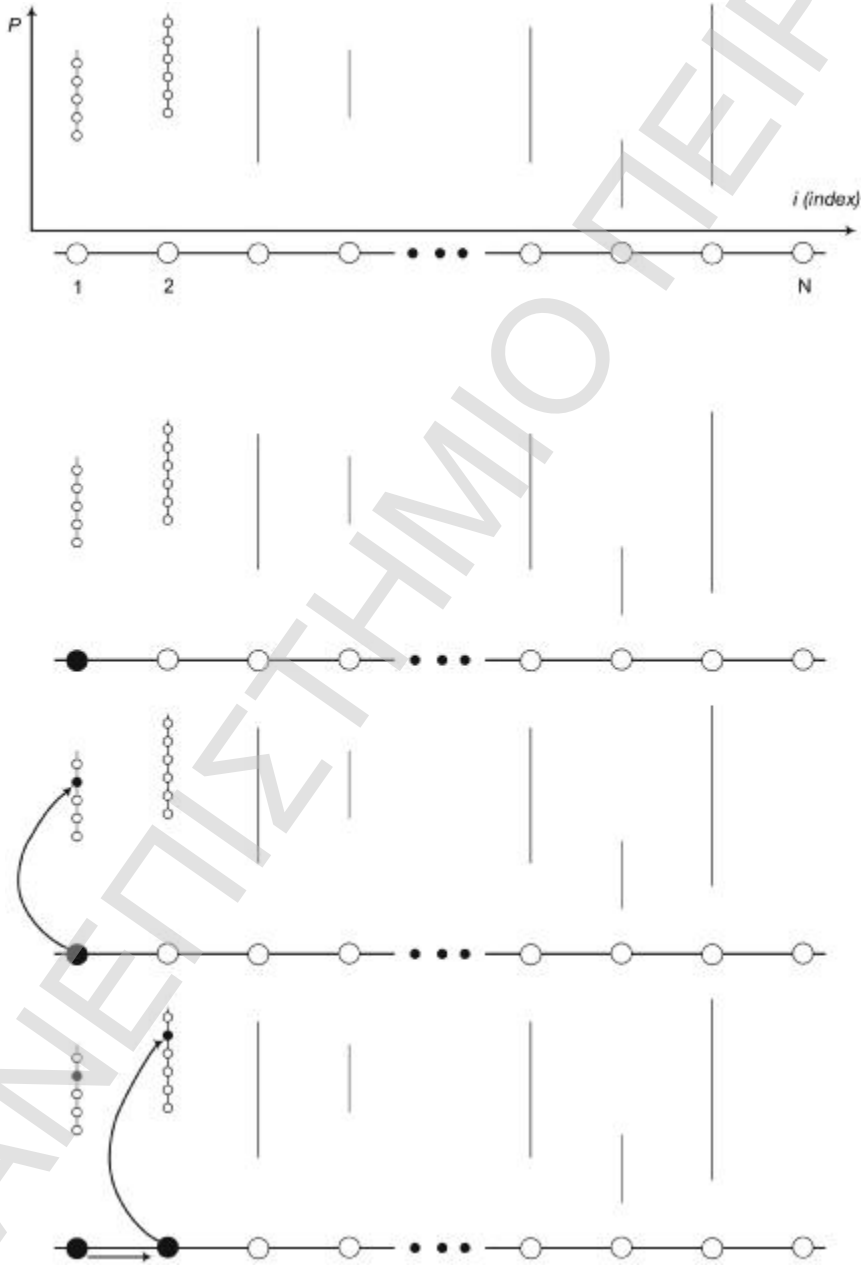
Στην περίπτωση υπέρβασης του μεγίστου ορίου παραγωγής ισχύος της γεννήτριας αναφοράς (στην οποία δίνεται ο δείκτης N) τότε ισχύει:

$$\text{dif} = P_N - P_{N,max}$$

Το συνολικό κόστος (9.1) υπολογίζεται συμπεριλαμβάνοντας και έναν παράγοντα ποινής $F_{tot,max}\text{dif}$. Στην αρχική λύση δεν φαίνεται να λαμβάνεται μέριμνα για την περίπτωση που η γεννήτρια αναφοράς υπολειτουργεί παραβιάζοντας το κάτω όριο ισχύος. Ένας άλλος παράγοντας ποινής μπορεί να είναι: $F_{tot,max}\text{dif}2$ όπου $\text{dif}2 = P_{N,min} - P_N$.

Ο αλγόριθμος λειτουργεί ως εξής: κάθε μυρμήγκι ξεκινάει από την πρώτη γεννήτρια και επιλέγει μια στάθμη ισχύος. Παρακάτω, θα αναλυθεί

πώς γίνεται αυτή η επιλογή. Στη συνέχεια κατευθύνεται στην επόμενη γεννήτρια και επιλέγει μια άλλη στάθμη ισχύος της, και αυτό επαναλαμβάνεται μέχρι να φτάσει στην τελευταία γεννήτρια η οποία είναι η γεννήτρια αναφοράς, υπεύθυνη για το ισοζύγιο ισχύος και η οποία παίρνει συνεχείς τιμές (σχήμα 9.1). Στο τέλος υπολογίζεται το συνολικό κόστος που χρησιμεύει για την αξιολόγηση της λύσης του προβλήματος ELD.



Σχήμα 1: Τρόπος διάσχισης των σταθμών ισχύων από τα μυρμήγκια.

Πριν προταθεί ο αλγόριθμος σε μορφή βημάτων και δοθεί ο ψευδοκώδικας (Παράρτημα ΙΙ) θα παρουσιαστεί το μαθηματικό μοντέλο του αλγορίθμου Max – Min Ant System (MMAS) το οποίο προσαρμόζεται κατάλληλα στα δεδομένα του προβλήματος ELD.

9.4 Μαθηματικό μοντέλο

Α) Κανόνας μετάβασης:

Ένα μυρμήγκι k που βρίσκεται στη γεννήτρια i επιλέγει μια στάθμη ισχύος j για αυτή:

$$j=\begin{cases}\max_{u\in J_i^k}\left\{\left[\tau_{iu}(t)\right]^{\alpha}\cdot\left[n_{iu}\right]^{\beta} & , \quad \alpha\nu \quad q\le q_0\\ J & , \quad \alpha\nu \quad q>q_0\end{cases} \tag{9.5}$$

όπου,

q είναι μια τυχαία μεταβλητή κατανεμημένη στο διάστημα $[0,1]$,

q_0 είναι μια καθοριζόμενη από το χρήστη παράμετρος με τιμή στο διάστημα $[0,1]$,

J είναι μία στάθμη ισχύος που επιλέγεται τυχαία σύμφωνα με την παρακάτω εξίσωση:

$$p_{ij}^k(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[n_{ij}\right]^{\beta}}{\sum_{l\in J_i^k}\left[\tau_{il}(t)\right]^{\alpha}\cdot\left[n_{il}\right]^{\beta}}\ \alpha\nu\ j\in J_i^k \tag{9.6}$$

J_i^k είναι η λίστα με τις στάθμες ισχύος που αντιστοιχεί στη γεννήτρια i.

n_{ij} είναι η ορατότητα.

Στο κλασικό πρόβλημα του περιοδεύοντος πωλητή, η ορατότητα ορίζεται

σαν αντίστροφο της απόστασης d_{ij} μεταξύ δύο πόλεων (i, j): $n_{ij} = \frac{1}{d_{ij}}$

Στην παρούσα υλοποίηση θα χρησιμοποιηθεί το αντίστροφο του κόστους για τη συγκεκριμένη στάθμη ισχύος: $n_{ij} = \frac{1}{F_i\left(P_{i,j}\right)}$

Β) Κανόνας ανανέωσης φερομόνης

$\tau_{ij}(t)$ είναι η ποσότητα φερομόνης στο τόξο που συνδέει κάθε γεννήτρια (εκτός της γεννήτριας αναφοράς) με τη στάθμη ισχύος (στάθμη λειτουργίας) της. Για την εφαρμογή του αλγορίθμου Max – Min Ant System στο πρόβλημα ELD απαιτούνται ορισμένες επισημάνσεις σχετικά με τον τρόπο που υπολογίζεται η φερομόνη:

1. Ανανέωση της φερομόνης πραγματοποιείται μόνον από ένα μυρμήγκι σε κάθε επανάληψη, είτε αυτό που έχει βρει τη συνολική βέλτιστη λύση είτε αυτό που βρίσκει την καλύτερη λύση σε κάθε επανάληψη. Αυτοί οι δύο μηχανισμοί αναζήτησης λύσης μπορούν να συνδυαστούν.

2. Η εξίσωση ανανέωσης φερομόνης (Pheromone update rule) είναι:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (9.7)$$

όπου:

$\tau_{ij}(t)$ είναι η ποσότητα φερομόνης που βρίσκεται στο τόξο που συνδέει κάθε γεννήτρια (εκτός της γεννήτριας αναφοράς) με τη στάθμη ισχύος (στάθμη λειτουργίας).

Κατά τη χρονική στιγμή $t = 0$ (πρώτη επανάληψη) η φερομόνη τίθεται σε μια αρχική τιμή ίση με τ_0 πολύ μεγάλη. Ιδανικά θα έπρεπε να τεθεί ίση με τ_{max}. Επειδή ο αλγόριθμος δεν έχει ξεκινήσει ακόμα, ορίζεται μια πολύ μεγάλη τιμή και κατά τη χρονική στιγμή $t = 1$ γίνεται $\tau_{ij}(1) = \tau_{max}$ (εκτός αν η τ_{max} αποδειχτεί μεγαλύτερη από την εκτίμηση του τ_0).

$$\Delta\tau_{ij}(t)\begin{cases} 1/\text{Cost}^{*}(t) & , \quad \text{αν} \quad (i,j)\in T^{*}(t) \\ 0 & , \quad \text{αν} \quad (i,j)\notin T^{*}(t) \end{cases} \tag{9.8}$$

όπου

$\text{Cost}^{*}(t)$ είναι το ελάχιστο κόστος μέχρι την παρούσα χρονική στιγμή t, $(\text{Cost}_{\text{best}}(t))$, είτε της καλύτερης λύσης κατά την επανάληψη t $(\text{Cost}_{\text{iter}}(t))$ και $T^{*}(t)$ η αντίστοιχη σειρά των στάθμεων ισχύος για κάθε γεννήτρια.

ρ είναι ο συντελεστής εξάτμισης της φερομόνης, $\rho\in[0,1]$

3. Μετά την ανανέωση της φερομόνης γίνεται έλεγχος αν είναι εντός των ορίων, $\tau_{\min}(t),\tau_{\max}(t)$. Τελικά, η ανανέωση της φερομόνης γίνεται με βάση την παρακάτω σχέση:

$$\tau_{ij}(t)\leftarrow\begin{cases} \tau_{\min}(t) & , \quad \text{αν} \quad \tau_{ij}(t)<\tau_{\min}(t) \\ \tau_{\max}(t) & , \quad \text{αν} \quad \tau_{ij}(t)>\tau_{\min}(t) \\ \tau_{ij}(t) & , \quad \text{αλλιώς} \end{cases} \tag{9.8}$$

4. Στον αλγόριθμο MMAS τοποθετούνται δύο όρια στα επίπεδα της φερομόνης.

   Το *άνω όριο* ορίζεται ως εξής:

$$\tau_{\max}(t)=\frac{1}{(1-\rho)\text{Cost}_{\text{opt}}}$$

   όπου Cost_{opt} είναι η πραγματική βέλτιστη λύση. Επειδή δεν είναι γνωστή από την αρχή, μπορεί να χρησιμοποιηθεί στη θέση του $\text{Cost}_{\text{best}}(t)$, που σημαίνει ότι το $\tau_{\max}$ θα μεταβάλλεται κάθε φορά που βρίσκεται μία καλύτερη λύση.

   Το κάτω όριο ορίζεται ως εξής:

$$\tau_{\min}(t)=\frac{\tau_{\max}(t)\left(1-\sqrt[n]{P_{\text{best}}}\right)}{(\lambda-1)\sqrt[n]{P_{\text{best}}}}$$

όπου P_{best} είναι η πιθανότητα δημιουργίας της συνολικής βέλτιστης λύσης. Η παράμετρος αυτή καθορίζεται από το χρήστη. Αν $P_{best}=1$ τότε $\tau_{min}=0$.

Αν P_{best} είναι πολύ μικρή τότε υπάρχει πιθανότητα να ισχύει $\tau_{min}(t)>\tau_{max}(t)$.

Όταν $\tau_{min}(t)=\tau_{max}(t)$ ο αλγόριθμος χρησιμοποιεί μόνον την ευρεστική πληροφορία (ορατότητα).

n είναι ο αριθμός των σημείων αποφάσεων (στάθμεων ισχύος). Σε κάθε στάθμη ισχύος το μυρμήγκι έχει να επιλέξει κατά μέσο όρο $\lambda\left(\lambda=\frac{n}{2}\right)$ τόξα που συνδέουν τη στάθμη ισχύος που βρίσκεται με τις υπόλοιπες, αντίστοιχα.

5. *Ομαλοποίηση της φερομόνης* (smoothing of pheromone)
(προαιρετικό βήμα)

Όταν ο αλγόριθμος έχει συγκλίνει υποδεικνύεται ο παρακάτω μηχανισμός ο οποίος προκαλεί αύξηση των επιπέδων φερομόνης ανάλογα με τη διαφορά τους από το $\tau_{max}(t)$:

$$\tau_{ij}^{*}(t)=\tau_{ij}+\delta\left(\tau_{max}(t)-\tau_{ij}(t)\right)$$

όπου
$\tau_{ij}^{*}(t)$ και $\tau_{ij}(t)$ είναι οι ποσότητες φερομόνης πριν και μετά την ομαλοποίηση, δ είναι μια παράμετρος που καθορίζεται από το χρήστη με $0\le\delta\le1$. Για $\delta=1$ ο μηχανισμός αυτός αντιστοιχεί σε μια επαναρχικοποίηση των επιπέδων φερομόνης. Για $\delta=0$ απενεργοποιείται ο μηχανισμός.

Ο μηχανισμός ομαλοποίησης φερομόνης χρησιμοποιείται κυρίως σε εκτελέσεις του αλγορίθμου με πολύ μεγάλο αριθμό επαναλήψεων.

9.5 Ο αλγόριθμος

Με τη μορφή των βημάτων ο αλγόριθμος που χρησιμοποιείται για τη λύση του προβλήματος ELD, έχει ως εξής:

Βήμα 1ο
Ορίζονται οι διακριτές στάθμες ισχύος για κάθε γεννήτρια καθώς και η γεννήτρια αναφοράς. Για κάθε γεννήτρια και κάθε στάθμη ισχύος υπολογίζεται το κόστος παραγωγής της συγκεκριμένης ποσότητας παραγωγής της συγκεκριμένης ποσότητας ισχύος και η ορατότητα

$$n_{ij}(t)\left(n_{ij} = \frac{1}{F_i\left(P_{i,j}\right)} \right)$$

Τοποθετείται η φερομόνη στην αρχική της τιμή – μια πολύ μεγάλη τιμή, σε όλα τα τόξα που συνδέουν κάθε γεννήτρια με τις αντίστοιχες στάθμες ισχύων της. Ορίζεται ο αριθμός των μυρμηγκιών και των επαναλήψεων.

Βήμα 2ο
Για κάθε μυρμήγκι και για κάθε γεννήτρια επιλέγεται η στάθμη ισχύος για τη συγκεκριμένη γεννήτρια με βάση τον κανόνα μετάβασης (transition rule) μέχρι που να έχει απομείνει η μηχανή αναφοράς.

Βήμα 3ο
Υπολογίζεται το κόστος για όλα τα μυρμήγκια με βάση τον καταμερισμό των ισχύων που πραγματοποίησαν και αποθηκεύεται το καλύτερο. Εδώ θα πρέπει να ελεγχθεί αν το καλύτερο αποτελεί μια αποδεκτή λύση, αν δεν παραβιάζει τα όρια ισχύων της γεννήτριας αναφοράς. Σε περίπτωση παραβίασης υπάρχουν δύο δυνατότητες:

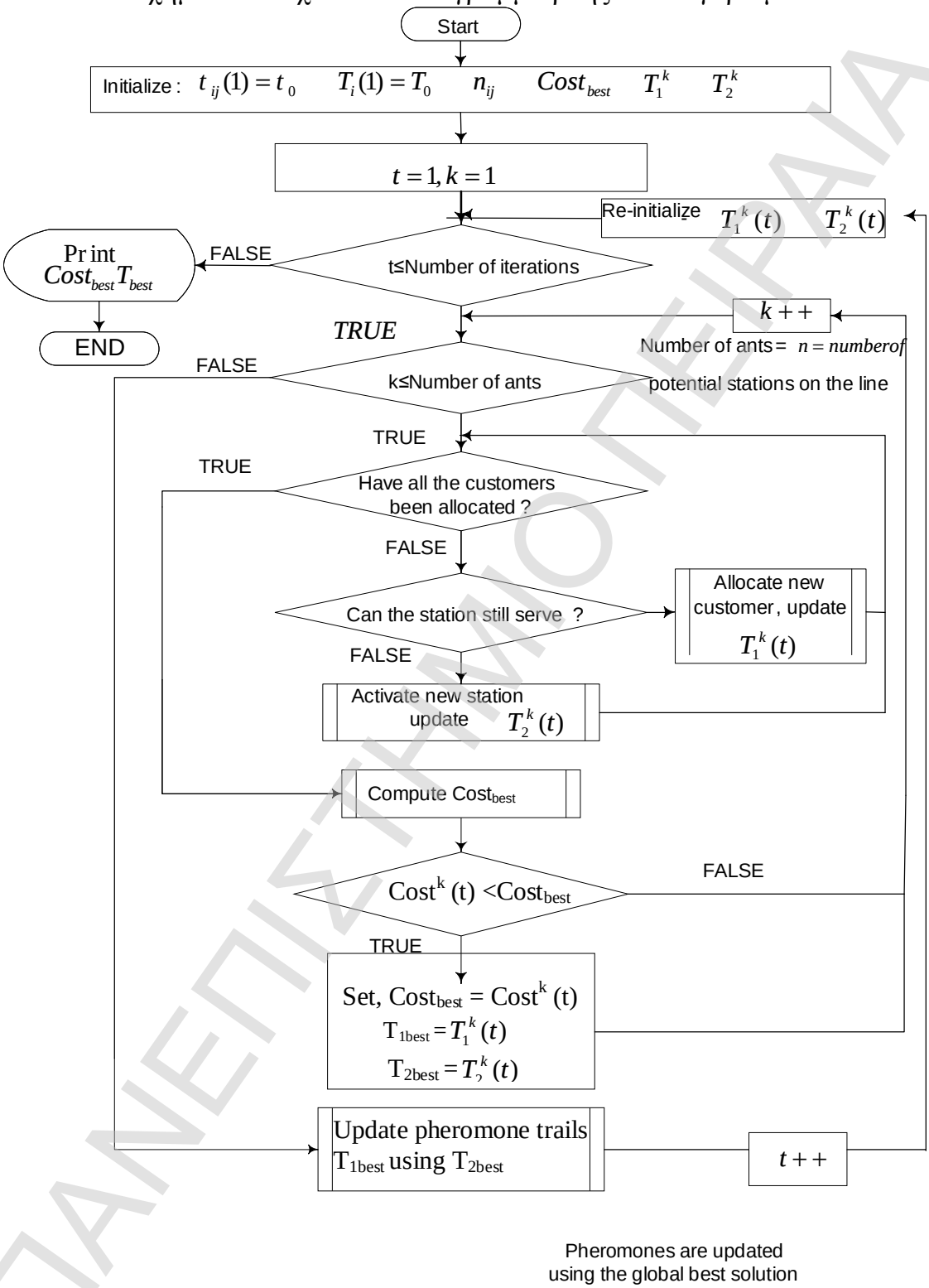
1. Απορρίπτεται η λύση και στη συνέχεια υλοποιείται η επόμενη επανάληψη.
2. Λαμβάνεται υπόψη αλλά σημειώνεται ότι δεν είναι αποδεκτή οπότε αν ο αλγόριθμος τερματίσει και έχει να προτείνει μια μη αποδεκτή λύση δεν εμφανίζεται και προτείνεται επανεκκίνηση με αλλαγή κάποιων παραμέτρων ή με μεταβολή της συνάρτησης κόστους ώστε οι μη αποδεκτές λύσεις να «αποθαρρύνονται» περισσότερο.

Βήμα 4ο
Ανανεώνεται η φερομόνη με βάση την εξίσωση (9.7) καθώς και τα όρια $\tau_{min}(t) - \tau_{max}(t)$.

Βήμα 5ο
Επαναλαμβάνεται η διαδικασία από το Βήμα 2 μέχρι να ολοκληρωθεί ένας συγκεκριμένος αριθμός επαναλήψεων.

Προαιρετικό βήμα
Όταν ο αλγόριθμος συγκλίνει εφαρμόζεται ο μηχανισμός ομαλοποίησης της φερομόνης.

Στο Σχήμα 9.2 δείχνεται το διάγραμμα ροής του αλγόριθμου.



Σχήμα 9.2: Διάγραμμα ροής του αλγορίθμου

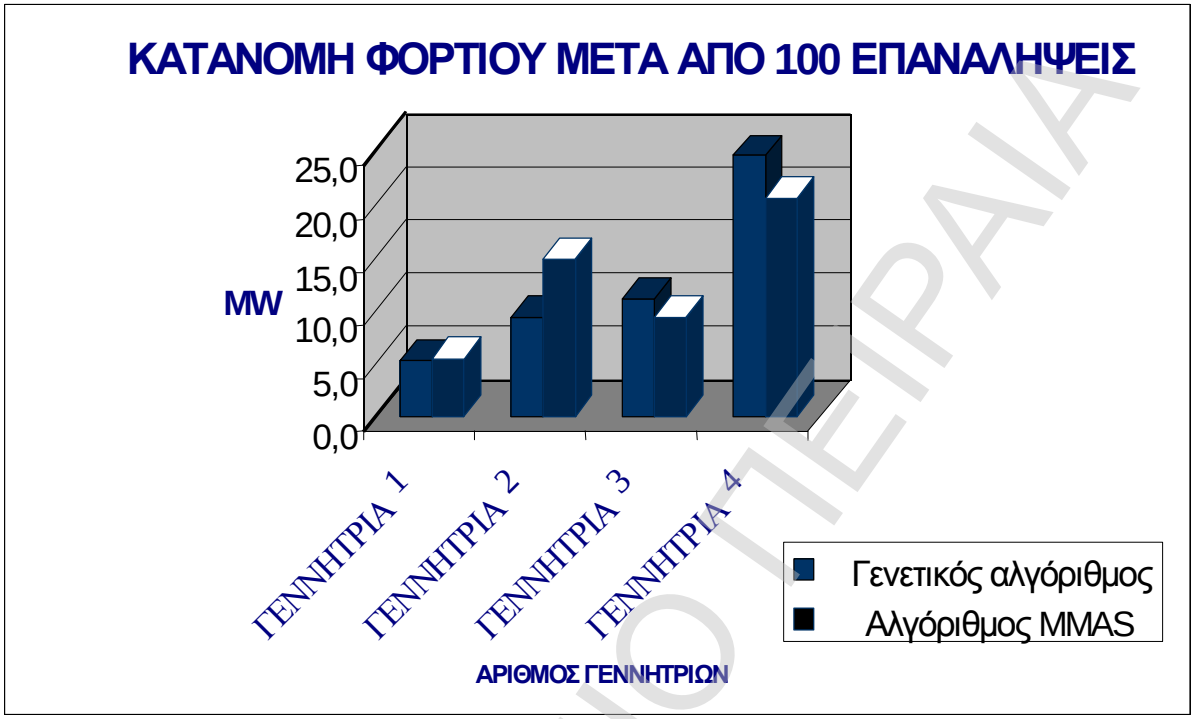
9.6 Εφαρμογή

Έστω ότι πρέπει να λυθεί το πρόβλημα οικονομικής κατανομής φορτίου τεσσάρων γεννητριών, όπου τα χαρακτηριστικά τους δείχνονται στον Πίνακα 1, εφαρμόζονται κατάλληλα τον αλγόριθμο MMAS. Οι συντελεστές a_i, b_i, c_i του πίνακα είναι οι συντελεστές κατανάλωσης καυσίμου, ενώ η στήλη Cost είναι το κόστος του καυσίμου. Οι γεννήτριες καλούνται να καλύψουν ζήτηση 50 MW. Η γεννήτρια 4 θεωρείται «γεννήτρια αναφοράς».

Πίνακας 1. Δεδομένα γεννητριών

i	Τύπος	$P_{i,min}$	$P_{i,max}$	a_i	b_i	c_i	Cost
1	Ατμοστρ	4.0	6.25	0.0	0.368	0.0	46.0
2	Φυσικό Αέριο	3.0	14.75	2.0938	0.24837	0.002270	123.0
3	Πετρελ.	3.0	12.28	0.3667	0.109	0.00425	46.0
4	Ατμοστρ	16.0	25.00	0.4053	0.2210	0.000643	46.0

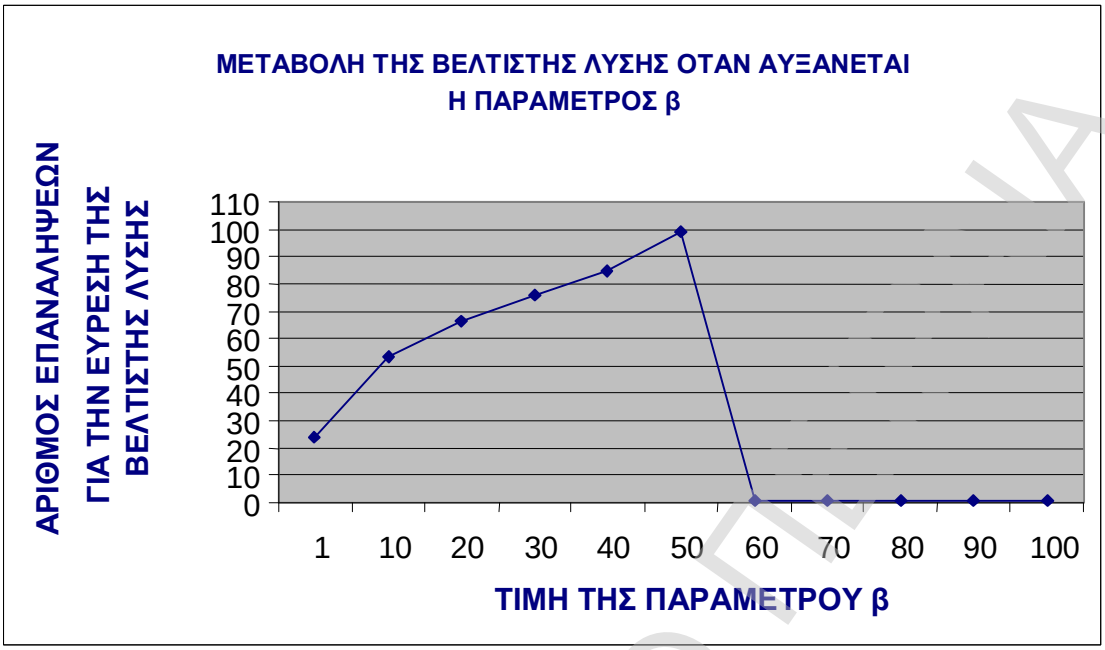
Στο σχ. 9.3 δείχνονται τα αποτελέσματα κατανομής φορτίου μετά από 100 επαναλήψεις και συγκρίνονται με τα αποτελέσματα εφαρμογής των γενετικών αλγορίθμων [Contaxis και άλλοι, 2001].



Σχήμα 9.3: Συγκριτικά αποτελέσματα γενετικών αλγορίθμων με τον αλγόριθμο MMAS μετά από 100 επαναλήψεις.

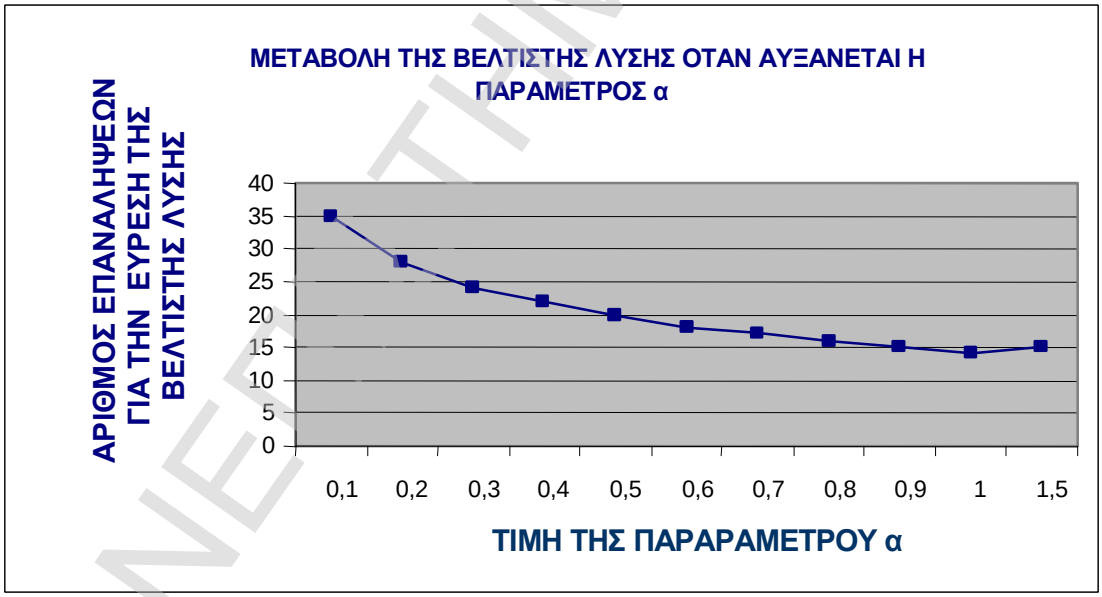
Στην παρούσα υλοποίηση ο αριθμός των γεννητριών είναι 4 ενώ οι τιμές των παραμέτρων q_0, p, α, β και τ_0 είναι 0.9, 0.1, 0.1, 2 και 1000.

Στο σχ. 9.4 δείχνεται η βέλτιστη λύση όταν μεταβάλλεται η παράμετρος β ενώ διατηρούνται σταθερές οι τιμές των παραμέτρων q_0, p, α και τ_0.



Σχήμα 9.4: Βέλτιστη λύση σε σχέση με την παράμετρο β.

Στο σχ. 9.5 δείχνεται η βέλτιστη λύση σε σχέση με τη μεταβολή της παραμέτρου α.



Σχήμα 9.5: Βέλτιστη λύση σε σχέση με την παράμετρο α.

Παρατηρείται ότι όσο η παράμετρος α αυξάνεται, τόσο γρηγορότερα βρίσκεται η βέλτιστη τιμή.

ΚΕΦΑΛΑΙΟ 10

ΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΩΡΟΘΕΤΗΣΗΣ – ΚΑΤΑΝΟΜΗΣ ΣΕ ΜΙΑ ΓΡΑΜΜΗ (CAPACITATED LOCATION – ALLOCATION PROBLEM ON A LINE) ΜΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ uP-ACS

10.1 Εισαγωγή

Το πρόβλημα χωροθέτησης – κατανομής, σχετίζεται με το συνδυασμένο προσδιορισμό της θέσης ενός αριθμού από νέες μονάδες και την κατανομή των ήδη υπαρχόντων μονάδων στις νέες μονάδες με σκοπό την εξυπηρέτηση. Το πρόβλημα αυτό μελετήθηκε για πρώτη φορά από τον Cooper (1863).

Για οικονομοτεχνικούς λόγους τα δίκτυα διανομής, συχνά, κατασκευάζονται με ιεραρχική δομή. Αρχικά, τα κανάλια διανομής προέρχονται από μια ή περισσότερες μονάδες. Στη συνέχεια διακλαδίζονται σε μικρότερα κανάλια με τρόπο παρόμοιο όπως το οργανικό αίμα και το νευρικό σύστημα στον ανθρώπινο οργανισμό.

Λόγω του μεγάλου κόστους, τα κανάλια μεγάλου επιπέδου, συχνά, δημιουργούνται σε ευθείες γραμμές, και από αυτά διακλαδώνονται χαμηλού επιπέδου κανάλια. Στην τοπολογία των δικτύων, η θέση των διακλαδωμένων σημείων και η κατανομή των υποδικτύων χαμηλού επιπέδου, σε αυτά, αποτελεί βασικό παράγοντα για τον προσδιορισμό των δαπανών υλοποίησης του συστήματος.

Τα κανάλια απαιτούν μεγάλο κόστος, λόγω και των ειδικών χαρακτηριστικών γνωρισμάτων τους όπως οι διαστάσεις τους, ταχύτητα, κλπ. Για τη μείωση των δαπανών τα κανάλια κατασκευάζονται, όπως αναφέρεται παραπάνω σε ευθείες γραμμές. Οι διακλαδώσεις βρίσκονται σε αυτές τις γραμμές. Οπότε, ο αριθμός των μονάδων, η θέση τους στις γραμμές και η κατανομή των αποστάσεών τους αποτελούν τα σημαντικά στοιχεία σχεδίασης των συστημάτων.

Σε αυτό το κεφάλαιο λύνεται το πρόβλημα χωροθέτησης – κατανομής σε μια γραμμή [Moshe Eben – Chaine και άλλοι, 2002], με τον αλγόριθμο

αποικίας μυρμηγκιών, uP-ACS, ο οποίος διαφοροποιείται πολύ από τον κλασικό αλγόριθμο ACS [Bonabean και άλλοι, 1999] τόσο ως προς τον πιθανοκρατικό κανόνα επιλογής όσο και ως προς τον κανόνα ανανέωσης φερομόνης.

10.2 Ορισμός του προβλήματος χωροθέτησης κατανομής (Capacitated Location – Allocation on A Line, CLAAL)

Το πρόβλημα χωροθέτησης – κατανομής είναι ένα πρόβλημα τοποθέτησης μονάδων σε μια ευθεία γραμμή καθώς και κατανομής των προορισμών προς εξυπηρέτηση σε μια από τις ήδη υπάρχουσες μονάδες.

Έστω ένα εργοστάσιο παραγωγής κάποιου προϊόντος, το οποίο μπορεί να έχει έναν αριθμό παραρτημάτων, όπου το καθένα από αυτά παράγει τον ίδιο τύπο προϊόντος. Με τον όρο μονάδα εννοείται το κάθε ένα από αυτά τα παραρτήματα.

Έστω ένας πελάτης που έχει παραγγείλει ένα προϊόν. Με τον όρο προορισμός εννοείται ο πελάτης.

Έστω κάποιες μονάδες και κάποιοι προορισμοί. Τότε πρέπει μόνο μια από τις μονάδες (εργοστάσιο), να παράγει το προϊόν και να παραδοθεί στον προορισμό (πελάτη). Με τον όρο κατανομή εννοείται η μονάδα που θα αναλάβει να εκτελέσει την παραγγελία.

Κάθε μονάδα μπορεί να παράγει ή να παρέχει ορισμένο αριθμό προϊόντων, ή υπηρεσιών, ο οποίος είναι γνωστός και ίδιος για όλες τις μονάδες. Αυτό αποτελεί τη χωρητικότητα της μονάδας. Σε καμία περίπτωση δεν μπορεί οι προορισμοί που αντιστοιχίζονται σε μια μονάδα να είναι περισσότεροι από τη χωρητικότητά της.

Δίνεται μια ευθεία γραμμή, στην οποία πρόκειται να κατασκευαστεί ένα πρωτεύον κανάλι διανομής, οι θέσεις και τα θετικά βάρη ενός πεπερασμένου συνόλου από n προορισμούς, που μπορεί να βρίσκονται οπουδήποτε πάνω στο επίπεδο, x, y, το κόστος τοποθέτησης (προμήθειας και εγκατάστασης) και η χωρητικότητα κάθε μονάδας.

Υποτίθεται ότι το κόστος λειτουργίας, που συσχετίζεται με τη μονάδα σε μια ορισμένη θέση και η απόσταση που κατανέμεται σε αυτή, είναι το σταθμισμένο άθροισμα των αποστάσεων μεταξύ μιας μονάδας και των κατανεμημένων προορισμών.

Προσδιορίστε τον αριθμό των μονάδων, τις θέσεις τους, πάνω στη γραμμή του πρωτεύοντος καναλιού και την κατανομή των αποστάσεων προς αυτές. Κάθε προορισμός πρέπει να είναι κατανεμημένος σε μια τουλάχιστον μονάδα, ενώ ο αριθμός των κατανεμημένων προορισμών σε κάθε μονάδα να μην υπερβαίνει τη χωρητικότητα της μονάδας και το συνολικό κόστος τοποθέτησης και λειτουργίας να είναι ελαχιστοποιημένο.

Το μαθηματικό μοντέλο ελαχιστοποίησης του προβλήματος CLAAL είναι:.

$$\min Z = \sum_{j=1}^{n}\left[Fy_i + \sum_{i=1}^{n} w_i d\left(x_j,(a_i,b_i)\right) z_{i,j} \right] \tag{10.1}$$

με τους περιορισμούς:

$$\sum_{j=1}^{n} z_{i,j} = 1, \qquad i = 1,\mathbf{K},n \tag{10.2}$$

$$\sum_{i=1}^{n} z_{i,j} \le K y_i, \qquad j = 1,\mathbf{K},n \tag{10.3}$$

$$y_j = 0/1, \qquad j = 1,\mathbf{K},n \tag{10.4}$$

$$z_{i,j} = 0/1, \qquad i = 1,\mathbf{K},n, \quad j = 1,\mathbf{K},n \tag{10.5}$$

Όπου η F δίνει το κόστος τοποθέτησης και η K δίνει το κόστος της χωρητικότητας της μονάδας.

Οι μεταβλητές y_j δείχνουν αν η μονάδα j έχει τοποθετηθεί $\left(y_j = 1\right)$ ή όχι $\left(y_j = 0\right)$ ενώ οι μεταβλητές $z_{i,j}$, οι μεταβλητές κατανομής, δείχνουν σε ποια μονάδα ο προορισμός i έχει κατανεμηθεί. Στην περίπτωση που η μονάδα j έχει τοποθετηθεί, μέχρι K προορισμοί μπορούν να συνδεθούν σε αυτή στο δεύτερο σύνολο περιορισμών, ενώ χρεώνεται και το κατάλληλο κόστος λειτουργίας συν F μονάδες κόστους τοποθέτησης στην αντικειμενική συνάρτηση. Σε αντίθετη περίπτωση κανένα κόστος δεν χρεώνεται, και πιο σημαντικά, κανένας προορισμός δεν μπορεί να συνδεθεί. Το πρώτο σύνολο περιορισμών εξασφαλίζει ότι κάθε προορισμός κατανέμεται σε μια μονάδα. Η συνάρτηση $d\left(x_j,(a_i,b_i)\right)$ μετράει την

απόσταση μεταξύ της μονάδας j, που βρίσκεται στη θέση x_j και τον προορισμό i στη θέση (a_i, b_i). Τα δεδομένα εισόδου είναι δοσμένα, έτσι ώστε ο άξονας x να συμπίπτει με το πρωτεύον κανάλι. Για ένα δοσμένο σύνολο θέσεων μονάδων, το σύνολο τοποθέτησης είναι προκαθορισμένο και η σταθμική απόσταση $w_i d(x_j,(a_i, b_i))$ μπορεί να υπολογισθεί για κάθε ζευγάρι (i, j). Υποδηλώνοντας αυτούς τους συντελεστές $c_{i,j}$ οι προορισμοί μπορούν να κατανεμηθούν σε μονάδες λύνοντας το πρόβλημα μεταφοράς που προκύπτει.

10.3 Μοντελοποίηση του προβλήματος

Για τη λύση του προβλήματος CLAAL χρησιμοποιήθηκε αλγόριθμος uP-ACS που στηρίζεται στη λειτουργία μιας αποικίας μυρμηγκιών. Συνήθως, οι αλγόριθμοι αποικίας μυρμηγκιών προορίζονται για τη λύση διακριτών προβλημάτων. Το πρόβλημα CLAAL αποτελείται από συνεχείς και διακριτές μεταβλητές. Γίνονται οι εξής παραδοχές:

- Σε κάθε θέση αντιστοιχεί μια και μόνον μια μονάδα.

- Έστω ευθεία γραμμή με n σταθμούς οι οποίοι ισαπέχουν. Αν και αναφέρεται ως ευθεία γραμμή, τελικά πρόκειται για ευθύγραμμο τμήμα το οποίο έχει δεδομένη αρχή και τέλος. Το τέλος, από γεωμετρικής άποψης δεν έχει νόημα να έχει τετμημένη μεγαλύτερη από τη μέγιστη τετμημένη των μονάδων – προορισμών.

Προκειμένου να διακριτοποιηθεί πλήρως το πρόβλημα CLAAL εκτός από τη φερομόνη που εναποθέτει το μυρμήγκι στο τόξο που ενώνει τη μονάδα με τον προορισμό, επιβάλλεται η τοποθέτηση φερομόνης και στη θέση που τοποθετείται η μονάδα.

Κανόνας μετάβασης 1 (transition rule 1):

Ο κανόνας μετάβασης 1 αφορά την κατανομή των προορισμών. Έστω ότι ένα μυρμήγκι k βρίσκεται σε μια μονάδα j, επιλέγει τον επόμενο προορισμό i με βάση τον κανόνα μετάβασης 1:

$$i = \begin{cases} \max_{u \notin I_1^k} \left\{ \left[\tau(i,u) \right] \cdot \left[n(i,u) \right]^{\beta} \right. & , \quad \text{αν} \quad q \le 0 \\ I & , \quad \text{αν} \quad q \ge 0 \end{cases} \tag{10.6}$$

όπου,

q είναι μια τυχαία μεταβλητή κατανεμημένη στο διάστημα $[0,1]$,

q_0 είναι μια καθορισμένη από το χρήστη παράμετρος με τιμή στο διάστημα $[0,1]$

I είναι ένας προορισμός που επιλέγεται τυχαία σύμφωνα με την παρακάτω εξίσωση:

$$p_{iJ}^k(t) = \frac{\tau_{ij}(t) \cdot \left[n_{ij}(t) \right]^{\beta}}{\sum_{l \notin J_1^k} \tau_{il}(t) \cdot \left[n_{il}(t) \right]^{\beta}} \quad \text{αν } j \in J_1^k \tag{10.7}$$

I_1^k είναι η λίστα με τους προορισμούς που εξυπηρετεί ήδη το μυρμήγκι

n_{ij} είναι το αντίστροφο της σταθμισμένης απόστασης μεταξύ μονάδας – προορισμού

$$n_{ij} = \frac{1}{w_j d_{ij}} \text{ (ορατότητα)}$$

$r_{ij}(t)$ είναι η ποσότητα φερομόνης που βρίσκεται στο τόξο που ενώνει μια μονάδα j με ένα προορισμό i κατά τη χρονική στιγμή $t = 0$ (πρώτη επανάληψη), η φερομόνη τίθεται σε μια αρχική τιμή r_0 πολύ μικρή.

Κανόνας μετάβασης 2 (transition rule 2):

Ο κανόνας μετάβασης 2 αφορά τη θέση της μονάδας. Έστω ότι το μυρμήγκι k βρίσκεται σε μια μονάδα η οποία έχει καλύψει το μέγιστο αριθμό προορισμών. Τότε επιλέγεται η επόμενη μονάδα p που πρέπει να

«ενεργοποιηθεί» με βάση τον κανόνα μετάβασης 2:

$$p = \begin{cases} \max_{u \notin J_2^k} \{T_u(t) & , \quad \alpha\nu \quad r \le r_0 \\ P & , \quad \alpha\nu \quad r > r_0 \end{cases} \tag{10.8}$$

όπου,

r είναι μια τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο διάστημα $[0,1]$,

r_0 είναι μια καθορισμένη από το χρήστη παράμετρος με τιμή στο διάστημα $[0,1]$,

p είναι μια μονάδα που επιλέγεται τυχαία σύμφωνα με την παρακάτω εξίσωση:

$$Q_p^k(t) = \frac{T_l(t)}{\sum_{l \notin J_2^k} T_l(t)} \quad \alpha\nu \quad P \notin J_2^k \tag{10.9}$$

όπου,

I_2^k είναι οι ενεργοποιημένες μονάδες

$T_i(t)$ είναι η ποσότητα φερομόνης που τοποθετείται σε κάθε μονάδα. Κατά τη χρονική στιγμή $t = 0$ (πρώτη επανάληψη), η φερομόνη τίθεται σε μια αρχική τιμή τ_0 πολύ μικρή.

Κανόνες ανανέωσης φερομόνης (Rules of pheromone replenishment)

Για το πρόβλημα CLAAL χρησιμοποιούνται δύο κανόνες ανανέωσης φερομόνης. Ο πρώτος κανόνας σχετίζεται με την αναζήτηση του καλύτερου συνδυασμού προορισμών (πελατών), που θα εξυπηρετούνται από μια μονάδα, έτσι ώστε να ελαχιστοποιείται το κόστος διανομής. Ο δεύτερος κανόνας σχετίζεται με την αναζήτηση του καλύτερου δυνατού συνδυασμού θέσης των μονάδων.

Κανόνας ανανέωσης 1:

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

$$\text{με} \qquad \Delta\tau_{ij}(t) = \begin{cases} 1/\text{Cost}_{best}(t), & \text{αν} \quad (i,j) \in T_{1best}(t) \\ 0 \quad , & \text{αν} \quad (i,j) \notin T_{1best}(t) \end{cases}$$

Κανόνας ανανέωσης 2:

$$T_i(t+1) = (1-\rho)T_i(t) + \rho\Delta T_i(t)$$

$$\text{με} \qquad \Delta\tau_{ij}(t) = \begin{cases} 1/\text{Cost}_{best}(t), & \text{αν} \quad (i,j) \in T_{2best}(t) \\ 0 \quad , & \text{αν} \quad (i,j) \notin T_{2best}(t) \end{cases}$$

όπου,

$\text{Cost}_{best}(t)$ είναι το ελάχιστο κόστος

$T_{1best}(t)$ είναι η λίστα των τόξων που συνδέουν τις μονάδες με τους προορισμούς

$T_{2best}(t)$ είναι η λίστα των μονάδων που χρησιμοποιήθηκαν για την υλοποίηση αυτής της βέλτιστης λύσης (διάνυσμα), με ρ $(0 \le \rho \le 1)$, ο συντελεστής εξάτμισης που μπορεί να επιλεγεί διαφορετικό για τους δύο κανόνες.

Εφ' όσον βρεθεί μια καλύτερη λύση από την ήδη υπάρχουσα, τότε αυτή γίνεται η νέα καλύτερη λύση και λαμβάνονται υπόψη οι τιμές $\text{Cost}_{best}(t)$, $T_{1best}(t)$ και $T_{2best}(t)$. $\text{Cost}_{best}(t)$, $T_{1best}(t)$ και $T_{2best}(t)$ αναφέρονται στην καλύτερη λύση που βρέθηκε μέχρι την επανάληψη t και όχι στην καλύτερη λύση που βρέθηκε στην επανάληψη t.

10.4 Βήματα του αλγορίθμου uP-ACS

Τα βήματα του αλγορίθμου uP-ACS είναι τα εξής:

Βήμα 1^{ο}
Τοποθετείται η πρώτη φερομόνη στην αρχική της τιμή για όλα τα μονοπάτια που συνδέουν τις μονάδες με τους προορισμούς.
Τοποθετείται και η δεύτερη φερομόνη που αντιστοιχεί σε κάθε μονάδα ίση με την αρχική τους τιμή.
Υπολογίζεται το σύνολο των σταθμισμένων αποστάσεων μεταξύ μονάδων και προορισμών.
Τοποθετείται σε κάθε μονάδα ένα μυρμήγκι.

Βήμα 2^{ο}
Σε κάθε μυρμήγκι ανατίθενται ελεύθεροι προορισμοί μέχρι να καλυφθεί η μέγιστη χωρητικότητα (μέχρι να «γεμίσει») ή μέχρι να εξυπηρετηθούν όλοι οι προορισμοί. Εάν έχουν τελειώσει όλοι οι προορισμοί τότε επιλέγεται το Βήμα 4 διαφορετικά επιλέγεται το Βήμα 3.

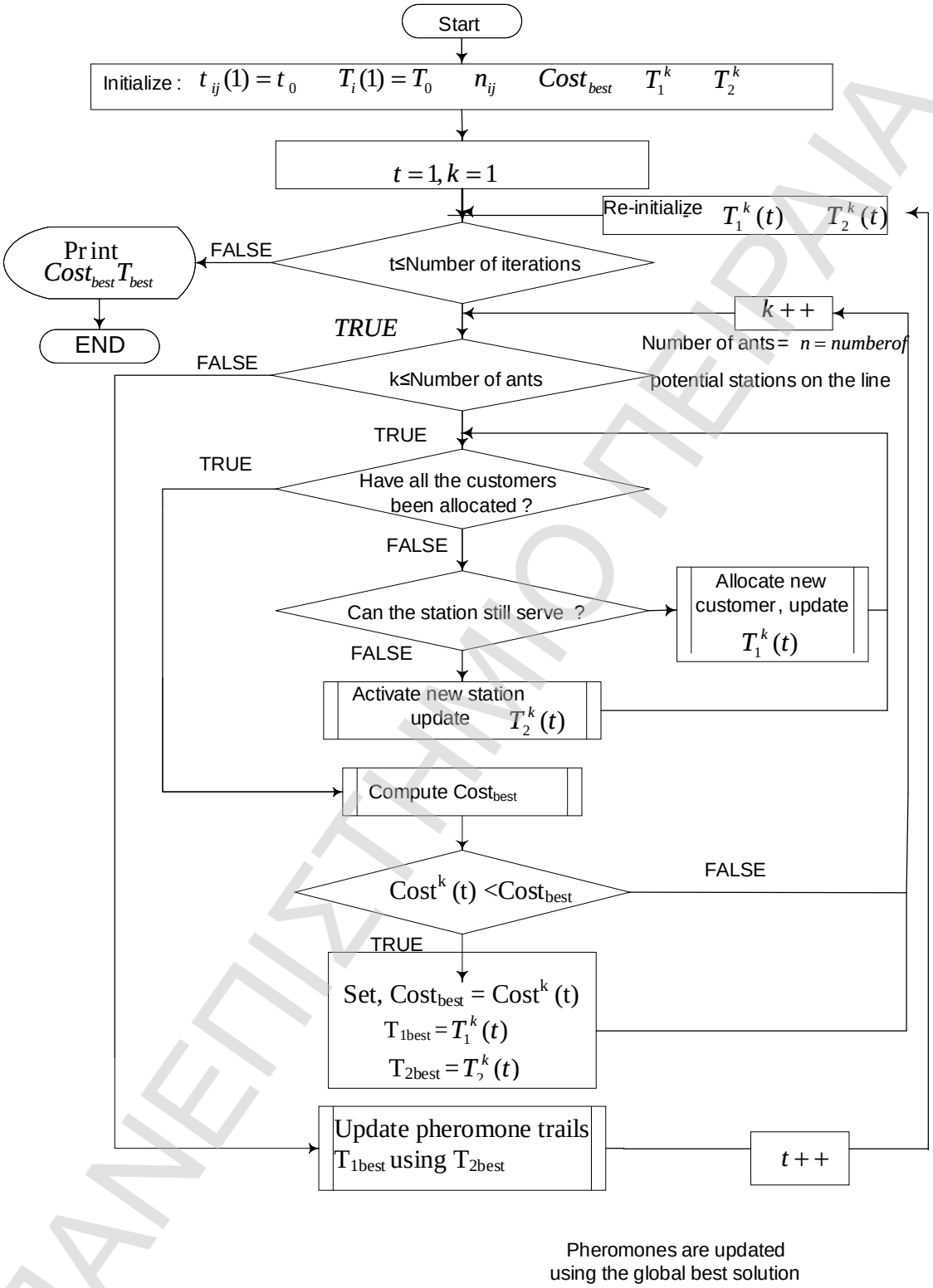
Βήμα 3^{ο}
Επιλέγεται η μονάδα που θα ενεργοποιηθεί με βάση τον κανόνα μετάβασης 2. γίνεται επιστροφή στο Βήμα 2 (για την ανάθεση προορισμών στη νέα μονάδα).

Βήμα 4^{ο}
Υπολογίζεται το κόστος για όλα τα μυρμήγκια και αποθηκεύεται το καλύτερο.

Βήμα 5^{ο}
Ανανεώνονται οι φερομόνες με βάση τους αντίστοιχους κανόνες.

Βήμα 6^{ο}
Επαναλαμβάνεται η διαδικασία από το Βήμα 2 μέχρι να ολοκληρωθεί ένας ορισμένος αριθμός επαναλήψεων.

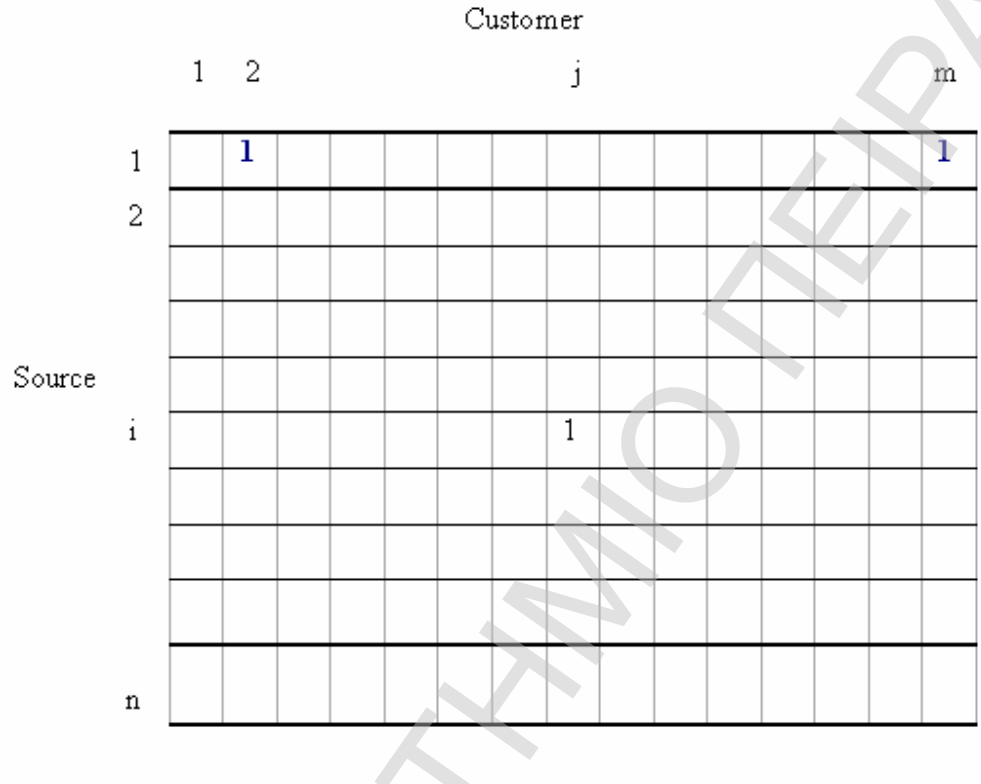
Ακολουθεί το διάγραμμα ροής του αλγορίθμου



Απαιτείται να κρατούνται τα ίχνη φερομόνης που τοποθετούν τα μυρμήγκια στην κάθε μονάδα μαζί με τους προορισμούς που προσδιορίζονται για κάθε μονάδα. Αυτό μπορεί να υλοποιηθεί μέσω ενός

διανύσματος ή ενός πίνακα. Μια ακριβής υλοποίηση γίνεται μέσω ενός πίνακα $n \times m$ (προορισμοί x μονάδες).

Όταν η γραμμή είναι κενή σημαίνει ότι στην αντίστοιχη θέση πάνω στη γραμμή, δεν υπάρχει μονάδα (σχ.2)



Σχήμα 2: Πίνακας προορισμών και μονάδων

Σε μια κάθετη στήλη θα υπάρχει μια μονάδα. Αυτό γιατί οι στήλες εκφράζουν τους προορισμούς. Έτσι, πρέπει να εξυπηρετούνται όλοι οι προορισμοί (μονάδες σε όλες τις στήλες) και ο κάθε προορισμός πρέπει να εξυπηρετείται από μια μόνο μονάδα.

Με βάση τον Πίνακα (tabu-list) του σχ. 2, η μονάδα που βρίσκεται στη θέση 1 (αρχή της γραμμής), εξυπηρετεί δύο προορισμούς (πελάτης-2 και πελάτης-n) ενώ η μονάδα που βρίσκεται στη θέση i εξυπηρετεί ένα μόνο προορισμό (πελάτης j).

10.5 Εφαρμογή

Ο αλγόριθμος uP-ACS εφαρμόστηκε για να ελαχιστοποιήσει τη συνάρτηση κόστους, εξίσωση (10.1), βρίσκοντας τον καλύτερο συνδυασμό μεταξύ προορισμών (πελατών) και μονάδων πάνω στην ευθεία γραμμή.

Παρατηρήθηκε ότι:

1. σε κάθε μονάδα δεν μπορούν να αντιστοιχίζονται περισσότεροι προορισμοί (πελάτες) από τη χωρητικότητά του,

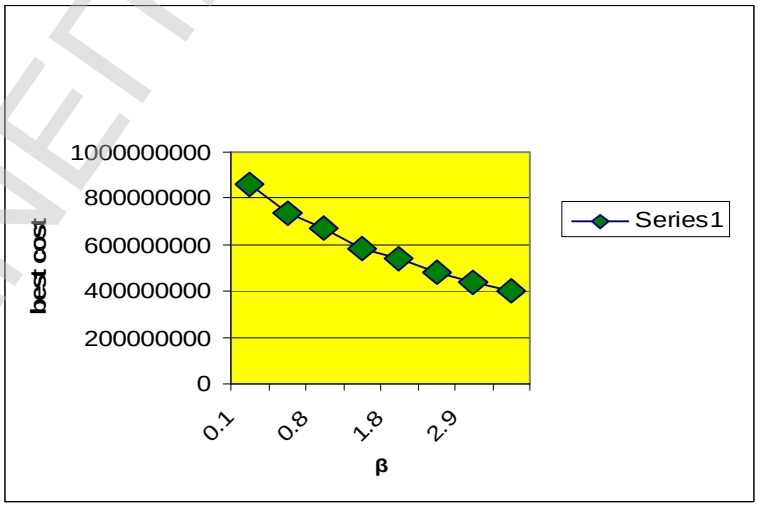
2. κάθε προορισμός (πελάτης) συνδέεται με μια μόνο μονάδα, και,

3. όλες οι μονάδες πρέπει να εξυπηρετούν.

Αρχικά το πρόβλημα υλοποιείται για τρεις σταθμούς με έξι (6) προορισμούς (πελάτες). Στον πίνακα 1 δείχνονται τα δεδομένα για τη δημιουργία της γραφικής παράστασης όπου παρουσιάζεται η μεταβολή της καλύτερης λύσης της συνάρτησης κόστους όταν μεταβάλλεται η παράμετρος β.

Πίνακας 1: Δεδομένα της εφαρμογής του uP-ACS

r_0	W_i	evaporation (ρ)	β	Station cost	iteration	Best cost
0,2	10,..,60	0,1	0,1	200	200	100
0,3	10,..,60	0,1	0,5	200	200	100
0,4	10,..,60	0,1	0,8	200	200	100
0,5	10,..,60	0,1	1,3	200	200	100
0,6	10,..,60	0,1	1,8	200	200	100
0,7	10,..,60	0,1	2	200	200	100
0,8	10,..,60	0,1	2,9	200	200	100
0,9	10,..,60	0,1	3,7	200	200	100



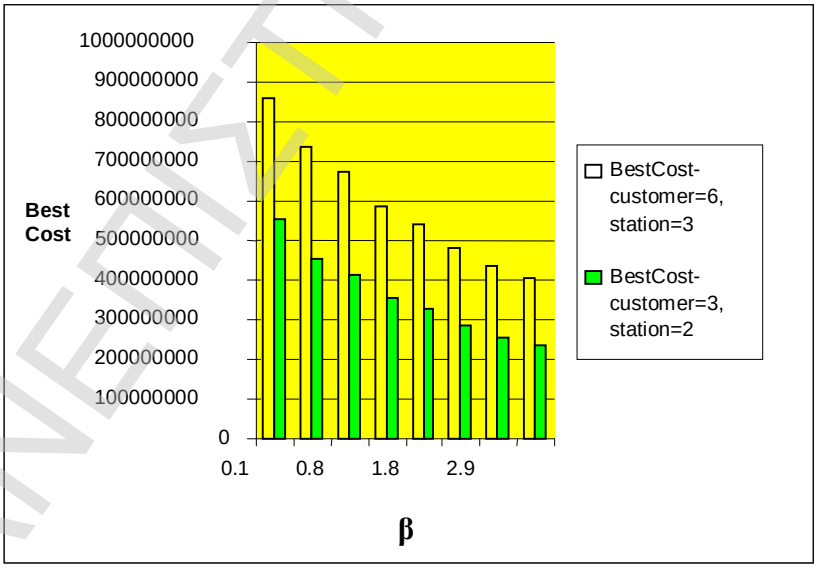
Σχήμα 3: Μεταβολή του καλύτερου κόστους με την παράμετρο b

Όταν ο αριθμός των μονάδων και των προορισμών είναι μικρός, η συνάρτηση κόστους θα είναι μικρή. Στο σχ. 3 δείχνεται ότι όταν η παράμετρος β αυξάνεται τότε η συνάρτηση κόστους ελαττώνεται.

Στον πίνακα 2 δίνονται τα δεδομένα τα οποία χρησιμοποιούνται για την κατασκευή της γραφικής παράστασης του σχ. 4, στην οποία συγκρίνεται το κόστος της συνάρτησης όταν οι μονάδες είναι τρεις (3) με έξι (6) προορισμούς (πελάτες) και όταν οι μονάδες είναι δύο (2) με τρεις (3) προορισμούς (πελάτες).

Πίνακας 2: Δεδομένα εφαρμογής του uP-ACS για τρεις και έξι μονάδες

β	Μονάδες = 6 station = 3	Μονάδες = 3 station = 2
01.	857800000	555700000
0.5	734800000	455700000
0.8	674800000	415700000
1.3	584800000	355700000
1.8	539800000	325700000
2.	479800000	285700000
2.9	434800000	255700000
3.7	404800000	235700000



Σχήμα 4: Σύγκριση του καλύτερου κόστους για τρεις και έξι μονάδες.

ΚΕΦΑΛΑΙΟ 11

ΒΕΛΤΙΣΤΗ ΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΓΡΑΜΜΙΚΗΣ ΤΟΠΟΘΕΤΗΣΗΣ ΜΗΧΑΝΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ

11.1 Εισαγωγή

Το σύστημα οργάνωσης της εργασίας σε «κυψέλες» μπορεί να εφαρμοστεί στην παραγωγή κατά παρτίδες βοηθώντας σημαντικά τη μαζική παραγωγή.

Η «κυψέλη» είναι μια ομάδα εργαζομένων με τα μηχανήματα και τα εργαλεία που χρησιμοποιούν και η ομάδα αυτή παράγει ένα ή περισσότερα είδη (ή και κατηγορίες προϊόντων ή τμημάτων του τελικού προϊόντος). Στη μαζική παραγωγή χρησιμοποιούνται τα έτοιμα προϊόντα σε μια συνεχή συναρμολόγηση. Η διαίρεση της εργασίας σε «κυψέλες» παρουσιάζει μεγάλη ελαστικότητα και ευκολία παραγωγής.

Όταν δημιουργηθούν μια ή περισσότερες «κυψέλες» σε μια εργοστασιακή μονάδα πρέπει να τοποθετηθούν οι μηχανές σε αυτές τις κυψέλες.

Υπάρχουν δύο βασικοί τύποι τοποθέτησης μηχανών μέσα σε μια «κυψέλη»:

i) γραμμική τοποθέτηση, και

ii) δικτυακή τοποθέτηση

Σε αυτό το κεφάλαιο παρουσιάζεται ένα γραμμικό μοντέλο, της ελάχιστης προς τα πίσω ροής (minimal backward-flow model), [Won, You-Dong, 1997] το οποίο θα βελτιστοποιηθεί μέσω αλγορίθμων της αποικίας μυρμηγκιών.

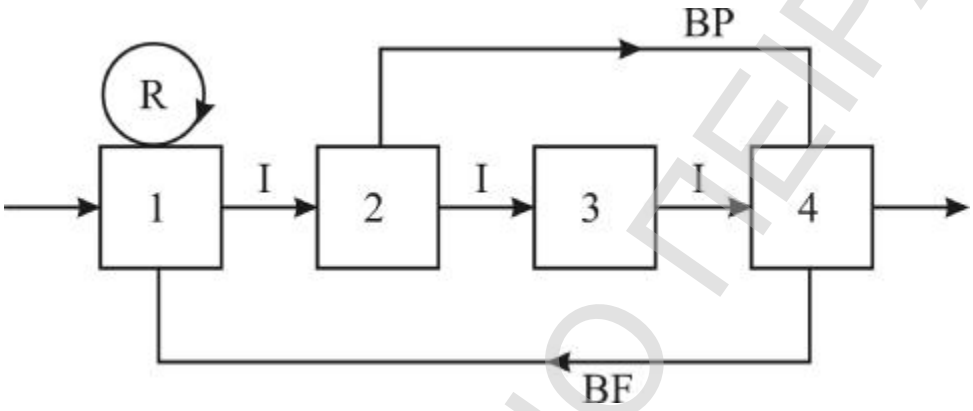
Σε μια γραμμική τοποθέτηση μηχανών υπάρχουν τέσσερις τύποι κινήσεων ροής (flow movements) σχ. 11-1 [Aneke Kon Carrier, 1986]:

1. επαναλαμβανόμενη, (R)

2. σε ακολουθία, (I),

3. σε παράκαμψη (BP), και

4. προς τα πίσω ροή (BF)



Σχ. 1: Τέσσερις τύποι κινήσεων ροής σε μια γραμμική τοποθέτηση μηχανών. R: επαναλαμβανόμενη, I: σε ακολουθία, BP: παράκαμψη, BF: προς τα πίσω ροή.

Η κίνηση προς τα πίσω ροής (Backward-flow) είναι η λιγότερο επιθυμητή γιατί περιπλέκει τη ροή εργασίας.

Ο βασικός σκοπός της ανάλυσης ροής είναι να ελαχιστοποιηθεί το συνολικό ποσό των προς τα πίσω ροής κινήσεων.

11.2 Διατύπωση του προβλήματος

11.2-1 Βιβλιογραφική ανασκόπηση της γραμμικής ανάλυσης χωροδιάταξης παραγωγής

Έχουν προταθεί διάφοροι αλγόριθμοι για τη γραμμική ανάλυση χωροδιάταξης παραγωγής. Οι περισσότεροι αλγόριθμοι επιτρέπουν διπλές μηχανές σε μια γραμμή ροής. Αρχικά, αυτός ο τύπος του προβλήματος μελετήθηκε από τον Carrie (1975). Η βασική του αρχή στηρίζεται στο σχεδιασμό μιας γραμμής που περιλαμβάνει έναν ικανό αριθμό σταθμών

εργασίας έτσι ώστε να μπορεί να προσαρμοστεί η ροή κάθε προϊόντος χωρίς να είναι απαραίτητη οποιαδήποτε προς τα πίσω ροή. Ο αλγόριθμος αποβάλλει τους αντιοικονομικούς σταθμούς εργασίας. Παράγονται πολλές λύσεις μετά από τη διεργασία αποβολών. Τελικά μέσω των προσομοιώσεων που υλοποιούνται στους υπολογιστές επιλέγεται η καλύτερη λύση.

Οι Aneke Kon Carrie (1986) παρουσίασαν μια μέθοδο που κατασκευάζει μια γραμμή ροής από τα δύο άκρα της γραμμής. Αυτή η μέθοδος αποδίδει καλύτερα σε μια κατάσταση όπου τα προϊόντα έχουν παρόμοιες απαιτήσεις των σταθμών εργασίας στα αρχικά και τα μεταγενέστερα στάδια επεξεργασίας τους. Αυτή η μέθοδος δεν περιλαμβάνει τη διαδικασία αποβολής για τους αντιοικονομικούς σταθμούς εργασίας.

Ο Lee (1991) πρότεινε μια μέθοδο που υλοποιεί μια γραμμική ροή βασισμένη στις ομοιότητες ακολουθίας των προϊόντων. Αυτός ο αλγόριθμος αρχίζει με την επιλογή του προϊόντος που έχει την υψηλότερη ομοιότητα ακολουθίας με τη γραμμή ακολουθίας που κατασκευάστηκε μέχρι τώρα. Στη συνέχεια τροποποιεί τη γραμμή ροής για να προσαρμόσει την ακολουθία του προσφάτου επιλεγμένου προϊόντος. Τέλος, εκτελείται η διαδικασία αποβολής των αντιοικονομικών σταθμών εργασίας.

11.3 Ελαχιστοποίηση του μοντέλου της προς τα πίσω ροής

11.3-1 Περιγραφή του μοντέλου

Η βασική αρχή που καθορίζει την ακολουθία των μηχανών είναι η ελαχιστοποίηση του συνολικού αριθμού των κινήσεων της προς τα πίσω ροής σε μια «κυψέλη» μιας εργοστασιακής μονάδας. Η προς τα πίσω ροή μπορεί να εκφραστεί μέσω ενός μοντέλου γραμμικού προγραμματισμού.

Για την ελαχιστοποίηση του μοντέλου της προς τα πίσω ροής υιοθετούνται οι παρακάτω παραδοχές [Won, You-Dong, 1997]

1. Μόνο μία μηχανή ενός τύπου τοποθετείται σε μια γραμμή ροής.

2. Το κόστος των υλικών των ροών είναι ανάλογο προς τον αριθμό των κομματιών που παράγονται και των αποστάσεων των ροών
3. Κάθε μηχανή θεωρείται σαν σημείο και η απόσταση μεταξύ διαδοχικών μηχανών είναι «1» (μοναδιαία απόσταση)

Η απόσταση μεταξύ του αρχικού σημείου εισόδου των προϊόντων, της κάθετης γραμμής αναφοράς vrl, και της πρώτης μηχανής στη σειρά θεωρείται, επίσης, σαν μοναδιαία απόσταση «1».

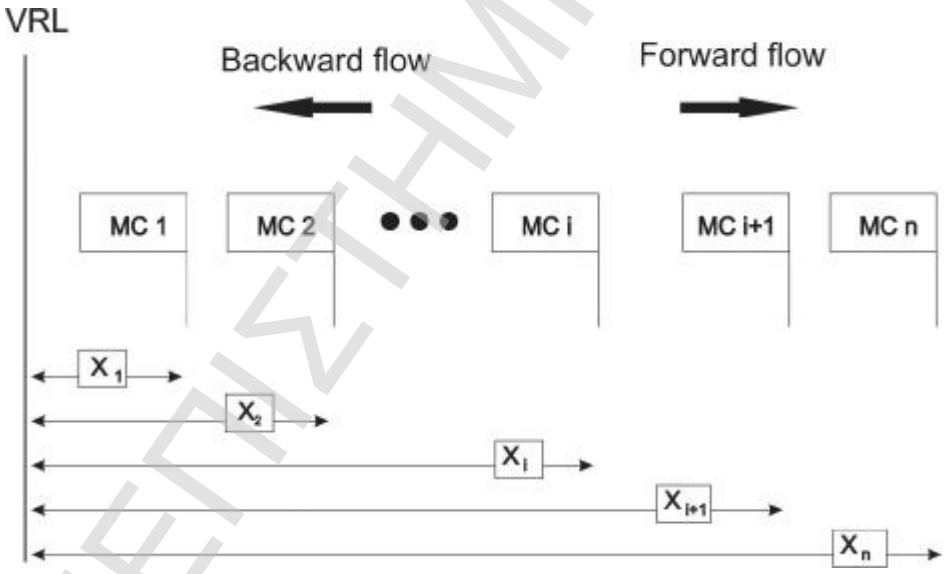
Στο Σχήμα 2 δίνεται, γραφικά, μια ακολουθία μηχανών. Στο μοντέλο της ελαχιστοποίησης προς τα πίσω ροής, παίρνονται υπόψη οι παρακάτω παράμετροι:

N: ο αριθμός μηχανών μιας «κυψέλης» σε μία εργοστασιακή μονάδα.

m: αριθμός προϊόντων που παράγονται σε μία «κυψέλη»

d_j: απαίτηση του προϊόντος j, $j = 1, 2, \mathbf{K}, n$

x_i: απόσταση μεταξύ της κάθετης γραμμής αναφοράς και της μηχανής i, $i = 1, 2, \mathbf{K}, n$



Σχήμα 2: Γραφική παράσταση της ακολουθίας μηχανών

Η μεταβλητή x_i είναι η μεταβλητή απόφασης που εκφράζει τη θέση της μηχανής i σε μια ακολουθία μηχανών.

Η αντικειμενική συνάρτηση που εκφράζει το μοντέλο ελαχιστοποίησης προς τα πίσω ροής είναι:

$$\min \sum_{h=1}^{n} \sum_{i=1}^{n} f_{hi} \left| x_n - x_i \right| \tag{11.1}$$

όπου:

$$h = 1, 2, \mathbf{K}, n$$

$$i = 1, 2, \mathbf{K}, n$$

$$j = 1, 2, \mathbf{K}, n$$

$$\left| x_h - x_i \right| \geq 1, \qquad h = 1, 2, \mathbf{K}, n \tag{11.2}$$

$$i = 1, 2, \mathbf{K}, n$$

$$h \neq i$$

$$x_i = 1, 2, \mathbf{K}, n \tag{11.3}$$

Η εξίσωση (11.1) εκφράζει τη συνολική απόσταση του «ταξιδιού» όλων των προϊόντων που παράγονται σε μια «κυψέλη» μιας εργοστασιακής μονάδας.

Ο περιορισμός (11.2) εξασφαλίζει ότι δεν επικαλύπτονται δύο μηχανές στη χωροδιάταξη. Η εξίσωση (11.3) εισάγει τις μεταβλητές απόφασης. Η συνολική απόσταση «ταξιδιού» αποτελείται από τις αποστάσεις της μπροστινής ροής (forward-flow) και τις αποστάσεις της προς τα πίσω ροής (backward-flow). Οι αποστάσεις της μπροστινής ροής δεν έχουν καμία επίδραση στη γραμμή ροής μιας «κυψέλης». Οπότε η αντικειμενική συνάρτηση εκφράζει την ελαχιστοποίηση του αθροίσματος των αποστάσεων της προς τα πίσω ροής ως εξής:

$$\min \sum_{h=1}^{n} \sum_{i=1}^{n} f_{hi} \cdot \max \left| x_n - x_i, 0 \right| \tag{11.4}$$

όπου:

f_{hi}: είναι μια συχνότητα βάρους παίρνοντας υπόψη την απαίτηση d_j, η οποία δίνεται:

$$f_{hi} = \begin{cases} \sum_{j=1}^{n} \alpha_{hij} d_i & , \quad h \neq i, \\ 0 & , \quad h = i \end{cases} \tag{11.5}$$

$$\text{με } \alpha_{hij} = \begin{cases} 1 & \text{αν η μηχανή i ακολουθεί} \\ & \text{τη διεργασία προϊόντος j} \\ 0 & \text{αλλιώς} \end{cases}$$

Η ελαχιστοποίηση του μοντέλου της προς τα πίσω ροής εκφράζεται από την εξίσωση (11.4) με τους περιορισμούς:

$$\left| x_h - x_i \right| \geq 1, \qquad h = 1,2,\mathbf{K},n \text{ , } i = h+1, h+2,\mathbf{K},n \tag{11.6}$$

$$x_i = 1,2,\mathbf{K},n$$

Το παραπάνω μοντέλο δεν είναι, ουσιαστικά, ένα μοντέλο γραμμικού προγραμματισμού γιατί περιλαμβάνει απόλυτες τιμές και τον περιορισμό (11.6). υπάρχει ανάγκη να μετασχηματιστεί το παραπάνω μοντέλο σε ένα ισοδύναμο $0-1$ μοντέλου ακέραιου προγραμματισμού, ορίζοντας:

$$z_{hi} = \max\left(x_h - x_i, 0 \right) \tag{11.7}$$

Είναι προφανές ότι:

$$z_{hi} \geq 0 \tag{11.8}$$

$$z_{hi} \geq x_h - x_i \tag{11.9}$$

$$x_h - x_i \geq 1 - M w_{hi} \tag{11.10}$$

$$-x_h + x_i \geq 1 - M\left(1 - w_{hi}\right) \tag{11.11}$$

όπου M ένας μεγάλος αριθμός και $w_{hi} = \begin{cases} 0 \\ 1 \end{cases}$

Τελικά, το $0-1$ ακέραιου προγραμματισμού μοντέλου

ελαχιστοποίησης της προς τα πίσω ροής είναι:

$$\min \sum_{h=1}^{n} \sum_{i=1}^{n} f_{hi} z_{hi} \qquad (11.12)$$

όπου:

$$x_h - x_i \geq 1 - M w_{hi}, \quad h = 1,2,\mathbf{K},n,\ i = h+1, h+2,\mathbf{K},n,\ h \neq i \qquad (11.13)$$

$$-x_h + x_i \geq M(1 - w_{hi}), \quad h = 1,2,\mathbf{K},n,\ i = h+1, h+2,\mathbf{K},n,\ h \neq i \qquad (11.14)$$

$$z_{hi} \geq x_h - x_i, \quad h = 1,2,\mathbf{K},n,\ i = h+1, h+2,\mathbf{K},n \qquad (11.15)$$

$$x_i = \sum_{j=1}^{n} j y_{ij}, \quad i = 1,2,\mathbf{K},n \qquad (11.16)$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad i = 1,2,\mathbf{K},n \qquad (11.17)$$

$$x_i, z_{hi} \geq 0, \quad w_{hi}, y_{ij} = 0 \text{ ή } 1 \qquad (11.18)$$

Προκύπτει ότι αν ο αριθμός των μηχανών σε μια «κυψέλη» είναι n, ο αριθμός των μεταβλητών και περιορισμών θα είναι $n(5n-1)/2$ και $2n^2$, αντίστοιχα.

11.4 Παρουσίαση του αλγορίθμου

Αναφέρθηκε στην προηγούμενη παράγραφο ότι το πρόβλημα δεν ορίζει κανένα κόστος / απόσταση στην περίπτωση της μπροστινής ροής για την παραγωγή ενός προϊόντος. Στο Σχήμα 2, αν σε μια διεργασία απαιτείται να χρησιμοποιηθεί η μηχανή 1 (MC1) και στη συνέχεια η μηχανή i (MCi) το κόστος / απόσταση ισούται με μηδέν και αν παρεμβάλλονται άλλες μηχανές ενδιάμεσα. Αντίθετα, αν σε μια διεργασία απαιτείται να χρησιμοποιηθεί η μηχανή 2(MC2) και στη συνέχεια η μηχανή 1(MC1) το κόστος θα είναι (1=απόσταση)·(ποσότητα παραγόμενου προϊόντος).

Με βάση τα παραπάνω θα υπολογιστούν μια σειρά «τοπικών»

αποστάσεων δηλαδή ένας πίνακας που θα έχει άμεση σχέση με την υλοποίηση του αλγορίθμου. Αν θεωρηθούν m μηχανές τότε θα κατασκευαστεί ένας πίνακας $m \times m$ κάθε στοιχείο του οποίου θα αποτελεί το κόστος / απόσταση για τη μετάβαση – τοποθέτηση μιας μηχανής στη θέση $i+1$ μετά από μια μηχανή που βρίσκεται στη θέση i.

Για τον υπολογισμό των στοιχείων αυτού του πίνακα θεωρείστε δύο (2) μόνο διεργασίες που χρησιμοποιούνται τρεις (3) μηχανές. Έστω δύο ορισμένες διεργασίες, για την παραγωγή δύο προϊόντων (Part 1, Part 2). Θα χρησιμοποιηθούν οι μηχανές 1 – 2 και 3 με την παρακάτω σειρά.

Πίνακας 1.
Παράδειγμα υπολογισμού στοιχείων του πίνακα

Part	Process	Demand
Part 1	1 – 2 – 3 – 2 – 3 – 1	10 pieces
Part 2	3 – 2 – 1 – 3 – 2- -3 – 1 – 2	15 pieces

Από τον Πίνακα 1, προκύπτει ένας πίνακας F, κόστους / αποστάσεων, γενικών κινήσεων

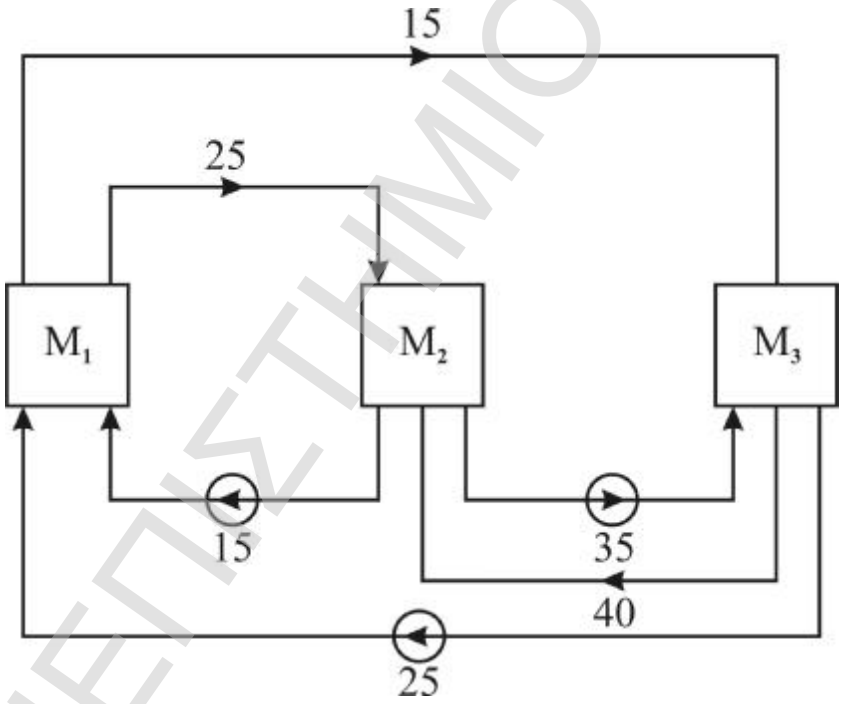
		στη $i+1$			
		M_1	M_2	M_3	
Από i	M_1	0	25	15	F
	M_2	15	0	35	
	M_3	25	40	0	

Ο Πίνακας που περιλαμβάνει τα κόστη / αποστάσεις που απαιτούνται για την προς τα πίσω ροής, για μια τοπική κίνηση, είναι ο ανάστροφος του παραπάνω. Οπότε ή υπολογίζεται κατευθείαν κινούμενος από το τέλος της διεργασίας προς την αρχή της ή από τον παραπάνω πίνακα με αναστροφή. Στο πρόγραμμα που υλοποιεί την παρούσα εργασία δίνεται ο ευθύς πίνακας και υπολογίζεται, αυτόματα, ο ανάστροφός του D.

Πίνακας 2.
Πίνακας κόστους / αποστάσεων του μοντέλου της προς τα πίσω ροής

		στο i			
		M_1	M_2	M_3	
	M_1	0	15	25	
Από $i+1i$	M_2	25	0	40	D
	M_3	15	35	0	

Στο Σχήμα 3 δείχνεται η τοποθέτηση των μηχανών 1 – 2 και 3 (M_1, M_2, M_3) με τα στοιχεία των πινάκων F και D αντίστοιχα.

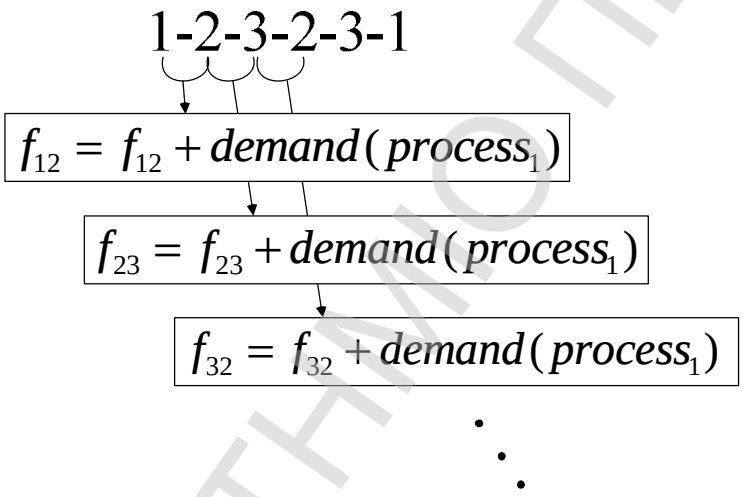


Σχήμα 3: Οι κύκλοι στις γραμμές ροής δείχνουν την ενεργοποίηση της προς τα πίσω ροής.

Η λογική του Πίνακα 2 είναι απλή. Αν σε μια διεργασία πρέπει μετά από τη μηχανή 1 να πάμε στη μηχανή 3 και στη χωροδιάταξη είναι τοποθετημένες διαδοχικά, τότε η προς τα πίσω ροή υπολογίζεται από τον πίνακα, και επίσης υπάρχει ο περιορισμός ότι δεν μπορεί να υπάρχει περισσότερο από μια φορά η ίδια μηχανή.

Ο πίνακας D μπορεί να υπολογιστεί, αλγοριθμικά, με δύο (2) loops. Αρχικοποιούμε όλα τα στοιχεία του πίνακα στο 0. Το εξωτερικό loop θα τρέχει από την πρώτη μέχρι την τελευταία διεργασία ενώ το εσωτερικό loop θα τρέχει την κάθε διεργασία και ο αριθμός της μηχανής που προηγείται αποτελεί τον πρώτο δείκτη του στοιχείου και ο αριθμός της μηχανής που έπεται τον δεύτερο δείκτη και προσθέτουμε τον αριθμό των προϊόντων που παράγει η αντίστοιχη διεργασία.

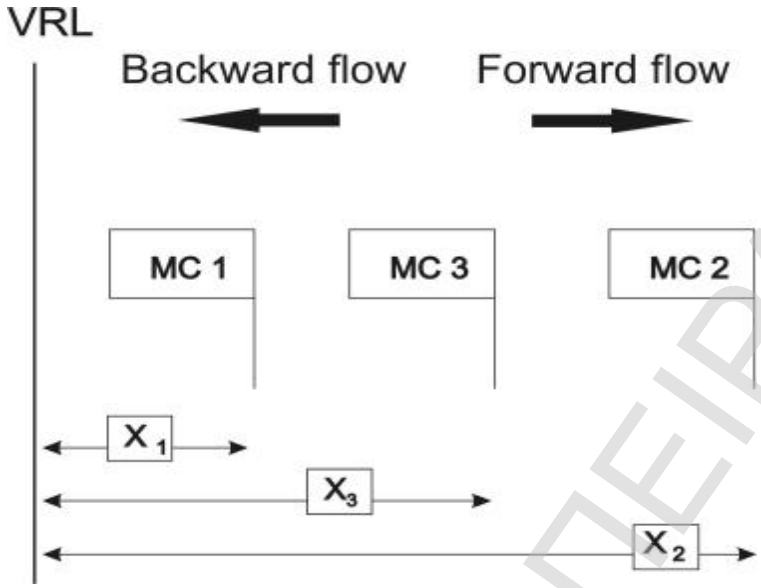
Έστω ότι υλοποιείται ο αλγόριθμος στην περίπτωση των δύο αναφερομένων διεργασιών, σχήμα 4.



Σχήμα 4: Υλοποίηση του αλγορίθμου για την ακολουθία μηχανών 1 – 2 – 3 – 2 – 3 – 1

Όταν υπολογιστεί ο πίνακας F και κατά συνέπεια ο πίνακας D δημιουργείται η αποικία μυρμηγκιών. Επειδή το κόστος / απόσταση όπως υπολογίζεται είναι τοπικό δεν ισούται με το συνολικό κόστος /απόσταση της χωροδιάταξης. Αυτό υπολογίζεται στο τέλος και όταν έχει ολοκληρώσει τη διαδρομή του.

Πράγματι, έστω ότι αποφασίζεται το σύνολο των τριών μηχανών να τοποθετηθούν σε μια γραμμή, όπως στο σχήμα 5.



Σχήμα 5: Ακολουθία μηχανών τοποθετημένων σε ευθεία γραμμή

Από τη χωροδιάταξη για την πρώτη διεργασία προκύπτει:

Πίνακας 3
Διαδικασία εύρεσης του κόστους / απόστασης για την πρώτη διεργασία

	Sign	Calculation	Result
VRL to M1	+	X1	1
M1 to M2	+	X2-X1	2
M2 to M3	-	X2-X3	1
M3 to M2	+	X2-X3	1
M2 to M3	-	X2-X3	1
M3 to M1	-	X3-X1	1
	Sum of negatives		-3

Cost for Process 1	Demand	Total Cost for Process 1
3	10	30

Από τον παραπάνω πίνακα με πρόσημο (+) δηλώνεται η μπροστινή ροή ενώ με το πρόσημο (–) η προς τα πίσω ροή. Αθροίζονται τα αρνητικά πρόσημα που εκφράζουν την προς τα πίσω ροή (η οποία είναι η ροή που δημιουργεί κόστη). Θεωρείστε το κόστος της $1^{ης}$ διεργασίας κατ' απόλυτο τιμή και πολλαπλασιάστε το με την απαίτηση της διεργασίας. Είναι φανερό ότι στον πίνακα 1 η διεργασία (1) απαιτεί την παραγωγή 10 προϊόντων. Το κόστος που αντιστοιχεί στη διεργασία (1) είναι 30.

Αυτή η δραστηριότητα επαναλαμβάνεται για κάθε διεργασία με στόχο την εύρεση του συνολικού κόστους (το οποίο θα είναι το άθροισμα όλων των διεργασιών μιας παραγωγής). Προτείνεται να χρησιμοποιηθούν τόσα μυρμήγκια όσες και οι μηχανές που θα αποτελέσουν τη γραμμή παραγωγής (αυτό σε αναλογία με τη χρήση ίσου αριθμού μυρμηγκιών με τις πόλεις που πρέπει να επισκεφτεί ο πωλητής στο TSP). Αν χρησιμοποιούνται m μηχανές τότε m θα είναι ο αριθμός των μυρμηγκιών της υλοποίησης του αλγορίθμου.

11.5 Απαιτήσεις του αλγορίθμου

Κανόνας μετάβασης

Αν το μυρμήγκι K βρίσκεται σε μια μηχανή i θα επιλέξει να επισκεφθεί την επόμενη μηχανή j σύμφωνα με τον κανόνα:

$$j = \begin{cases} \arg\max_{u \notin j_i^k} \left\{ \tau_{iu}(t) \cdot [n_{iu}]^{\beta} \right. & , \quad \text{αν} \quad q \le q_0 \\ 0 & , \quad \text{αν} \quad q > q_0 \end{cases} \tag{11.19}$$

όπου:

$\tau_{ij}(\tau)$: είναι η φερομόνη στο τόξο (i, j),

n_{ij}: η ορατότητα,

q: μια τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο διάστημα $[0,1]$,

q_0: είναι μια καθορισμένη από το χρήστη παράμετρος με τιμή στο διάστημα $[0,1]$,

j: είναι μια μηχανή που επιλέγεται τυχαία σύμφωνα με την παρακάτω εξίσωση:

$$P_{ij}^{k}(t)=\frac{\tau_{ij}(t)\cdot\left[n_{ij}\right]^{\beta}}{\sum_{l\notin j_i^k}\tau_{il}(t)\left[n_{ij}\right]^{\beta}}\ \text{αν}\ \ j\notin j_i^k \qquad (11.20)$$

j_i^k: είναι οι μηχανές στις οποίες είχε πάει το μυρμήγκι πριν από τη μηχανή i,

n_{ij}: είναι το αντίστροφο της απόστασης μεταξύ των δύο μηχανών (είναι τα στοιχεία του πίνακα D),

$\tau_{ij}(t)$: είναι η ποσότητα της φερομόνης που βρίσκεται στο τόξο που ενώνει δύο μηχανές.

Κατά τη χρονική στιγμή $t=0$ (πρώτη επανάληψη) η φερομόνη τίθεται σε αρχική τιμή τ_0, πάρα πολύ μικρή.

Ανανέωση φερομόνης

Ο μηχανισμός ανανέωσης της φερομόνης είναι:

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

όπου:

$$\Delta\tau_{ij}(t)=\begin{cases}1/Cost_{best}(t), & \text{αν}\quad (i,j)\in T_{best}(t)\\ 0\qquad\quad , & \text{αν}\quad (i,j)\notin T_{best}(t)\end{cases}$$

Με $Cost_{best}(t)$ θεωρείται το ελάχιστο κόστος όπως υπολογίζεται μετά το πέρας της τοποθέτησης των μηχανών ενώ $T_{best}(t)$ είναι η αντίστοιχη σειρά των μηχανών.

ρ με $0\le\rho\le1$ ο συντελεστής εξάτμισης της φερομόνης

Βήματα αλγορίθμου

Τα βήματα του αλγορίθμου είναι:

Βήμα 1^{ο}
Θέτουμε τη φερομόνη στην αρχική της τιμή σε όλες τις δυνατές αλληλουχίες μηχανών
Υπολογίζουμε τον Πίνακα D
Θέτουμε σε κάθε μηχανή ένα μυρμήγκι

Βήμα 2^{ο}
Για κάθε μυρμήγκι, επιλέγουμε την επόμενη μηχανή που θα επισκεφτεί (θα τοποθετήσει δηλαδή στην επόμενη θέση μπροστά από τη μηχανή στην οποία βρίσκεται) με βάση τον κανόνα που αναφέρθηκε πιο πάνω (transition rule) μέχρι να επισκεφτεί όλες τις μηχανές (να δημιουργήσει δηλαδή ένα layout).

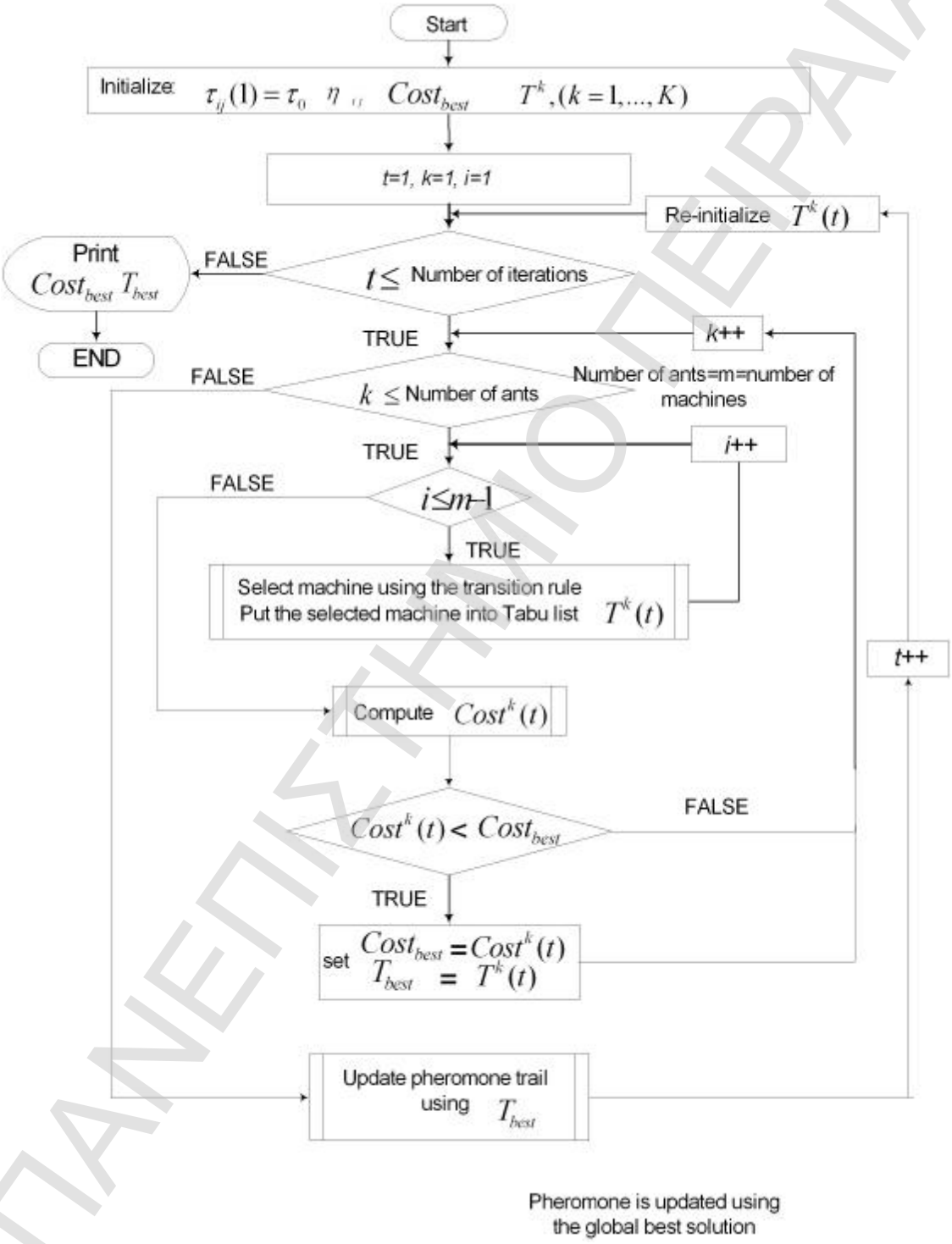
Βήμα 3^{ο}
Υπολογίζουμε το κόστος για όλα τα layout-διαδρομές και αποθηκεύουμε το καλύτερο.

Βήμα 4^{ο}
Ανανεώνουμε τη φερομόνη με βάση τον κανόνα που αναφέρθηκε πιο πάνω (Pheromone update rule)

Βήμα 5^{ο}
Επαναλαμβάνουμε τη διαδικασία από το βήμα 2 μέχρις ότου ολοκληρωθεί ένας συγκεκριμένος αριθμός επαναλήψεων ή ικανοποιηθεί κάποιο κριτήριο (π.χ. το κόστος από τα backward flow να πέσει κάτω από ένα κατώφλι που θέλουμε).

Στο Παράρτημα ΙΙΙ δίνεται ο ψευδοκώδικας του αλγορίθμου που εφαρμόζεται στο πρόβλημα.

Στο Σχήμα 6 δείχνεται το διάγραμμα ροής του αλγορίθμου



Σχήμα 6: Διάγραμμα ροής του αλγορίθμου

11.6 Εφαρμογή

Υποθέστε ότι πρέπει να παραχθούν πέντε προϊόντα P_1, P_2, P_3, P_4, P_5, με τρεις μηχανές 1, 2, 3 την πρώτη φορά και έξι μηχανές 1, 2, 3, 4, 5, 6 τη δεύτερη φορά. Οι απαιτήσεις και η ακολουθία των διεργασιών για κάθε προϊόν δίνεται στους πίνακες 2 και 3. Εισάγεται από το χρήστη το μέγεθος του προβλήματος, δηλαδή ο αριθμός των μηχανών (και τα αντίστοιχα μυρμήγκια) και διαδικασιών, το ποσό της φερομόνης (που συνήθως είναι σταθερό και θεωρείται πολύ μικρή ποσότητα: 0,1), το καλύτερο κόστος που επιθυμείται να επιτευχθεί για την παραγωγή των συγκεκριμένων προϊόντων καθώς και τον αριθμό των επαναλήψεων που πρέπει να υλοποιηθούν για την παραγωγή των προϊόντων που έχουν δηλωθεί.

Πίνακας 3
Δεδομένα για κάθε προϊόν, στην περίπτωση των τριών (3) μηχανών

Parts	Process	Quantity
P_1	$1 \rightarrow 3 \rightarrow 4$	10
P_2	$2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	20
P_3	$3 \rightarrow 4 \rightarrow 1$	15
P_4	$3 \rightarrow 4 \rightarrow 2 \rightarrow 1$	35
P_5	$3 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$	25

Πίνακας 4
Μέγεθος του προβλήματος

#	machine number	=	3
#	part number	=	5

Ένας σημαντικός παράγοντας στην εφαρμογή των αλγορίθμων αποικίας μυρμηγκιών, είναι η επιλογή της κατάλληλης τιμής της αρχικής ποσότητας της φερομόνης, η οποία διευκολύνει τις καλύτερες διαδρομές.

Συνήθως, η αρχική ποσότητα της φερομόνης είναι σταθερή και η τιμή της στην εφαρμογή θεωρείται μικρή: 0,1. Επίσης, στις εφαρμογές των αλγορίθμων αποικίας μυρμηγκιών σημαντική αξία έχουν παράμετροι που ρυθμίζονται από το χρήστη. Μια τέτοια παράμετρος είναι η β, που στην

παρούσα εφαρμογή θεωρείται ίση με 2.

Στον παρακάτω πίνακα δίνονται οι αποστάσεις μεταξύ των μηχανών (ή μπορεί να είναι ένας πίνακας κόστους).

$$\begin{array}{c} \\ \text{From i} \end{array} \begin{array}{c} \text{To i+1} \\ \begin{bmatrix} 0 & 25 & 15 \\ 15 & 0 & 35 \\ 25 & 40 & 0 \end{bmatrix} \end{array} \quad \text{F}$$

Παρακάτω δίνεται ο πίνακας με τις αποστάσεις της προς τα πίσω ροής όπως υπολογίζεται από το πρόγραμμα

$$\begin{array}{c} \\ \text{From i+1} \end{array} \begin{array}{c} \text{To i} \\ \begin{bmatrix} 0 & 15 & 25 \\ 25 & 0 & 40 \\ 15 & 35 & 0 \end{bmatrix} \end{array} \quad \text{D}$$

Στη συνέχεια υλοποιώντας το πρόγραμμα προκύπτουν τα αποτελέσματα της βέλτιστης ακολουθίας για κάθε προϊόν. Πίνακας 5: Σε αυτόν τον πίνακα δείχνονται τα καλύτερα κόστη για κάθε ροή προς τα πίσω και το μυρμήγκι – μηχανή που παράγει το καλύτερο κόστος.

Πίνακας 6
Αποτελέσματα για τρεις (3) μηχανές

Part	Best Cost	Machine-Ant	Machine Sequence
1	20	3	3 – 1 – 2
2	40	1	1 – 2 – 3
3	15	3	3 – 1 – 2
4	70	3	3 – 1 – 2
5		2	2 – 1

Στους πίνακες 6, 7 και 9 η προσομοίωση αφορά την παραγωγή πέντε (5) προϊόντων, με αριθμό μηχανών έξι (6) με τα αντίστοιχα αποτελέσματα

Πίνακας 6
Δεδομένα για κάθε προϊόν, στην περίπτωση έξι (6) μηχανών

Part	Process	Quantity
P_1	$1 \rightarrow 3 \rightarrow 4$	10
P_2	$2 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 2$	20
P_3	$3 \rightarrow 4 \rightarrow 1$	15
P_5	$3 \rightarrow 4 \rightarrow 2 \rightarrow 1$	35
P_5	$3 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 1$	25

Πίνακας 7
Μέγεθος του προβλήματος

#	machine number	=	6
#	part number	=	5

Πίνακας 8
Αποτελέσματα για έξι (6) μηχανές

Part	Best Cost	Machine-Ant	Machine Sequence
1	0	1	1 – 3 – 4 – 6 – 5 – 2
2	80	2	2 – 4 – 1 – 3 – 6 – 5
3	0	3	3 – 4 – 1 – 6 – 5 – 2
4	105	2	2 – 4 – 1 – 3 – 6 – 5
5	100	2	2 – 4 – 1 – 3 – 6 – 5

Όταν το πρόγραμμα τερματίσει δημιουργείται ένα αρχείο, το chainont.txt που δείχνεται πως υλοποιείται ο αλγόρι7θμος για κάθε μηχανή – μυρμήγκι και το αντίστοιχο κόστος καθώς και οι επαναλήψεις που ΄χουν γίνει, όπως δείχνεται στον πίνακα 9, για τρεις (3) μηχανές.

Πίνακας 9
Αποτελέσματα για τρεις (3) μηχανές

Iteration	Ant	Cost	Machine arrangement
1	1	140	1 – 2 – 3

1	2	105	2 – 1 – 3
1	3	70	3 – 1 – 2

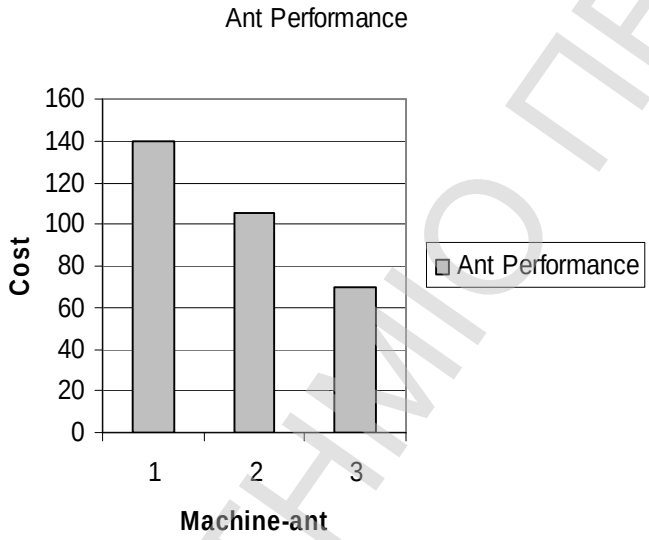
Τα αποτελέσματα σε αυτή την περίπτωση είναι:

Best Cost =70

Ant =3 (αντιστοιχεί στη μηχανή 3)

Machine Sequence = 3 – 1 – 2

Στο Σχήμα 7 δείχνεται η γραφική παράσταση ελαχιστοποίησης του κόστους, ενώ το καλύτερο κόστος το βρίσκει η μηχανή 3.



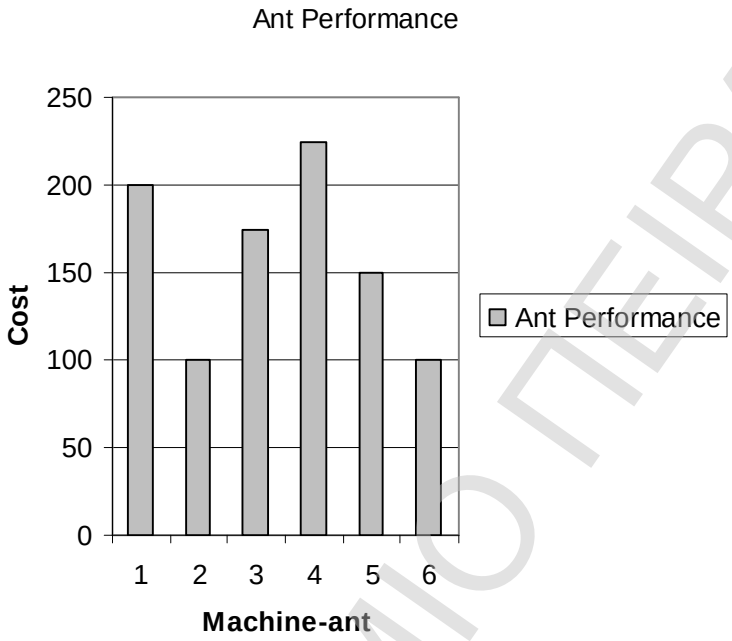
Σχήμα 7: Καλύτερο κόστος για 3 μηχανές

Είναι δυνατό το καλύτερο κόστος να μην το βρει η τελευταία μηχανή στη σειρά. Αυτό δείχνεται στον πίνακα 10.

Πίνακας 10
Αποτελέσματα για έξι (6) μηχανές

Iteration	Ant	Cost	Machine arrangement
1	1	200	1 – 3 – 4 – 6 – 5 – 2
1	2	100	2 – 4 – 1 – 3 – 6 – 5
1	3	175	3 – 4 – 1 – 6 – 5 – 2
1	4	225	4 – 1 – 3 – 6 – 5 – 2
1	5	150	5 – 1 – 3 – 4 – 6 – 2
1	6	100	6 – 1 – 3 – 4 – 2 – 5

Το καλύτερο κόστος το βρίσκει η μηχανή 2 και 5, όπως δείχνει στο σχήμα 8.



Σχήμα 8: Καλύτερο κόστος για έξι (6) μηχανές

Σαν αποτέλεσμα δηλώνεται αυτό της μηχανής 2 γιατί είναι πρώτη σε σειρά που βρίσκει το καλύτερο αποτέλεσμα

11.7 Συμπεράσματα

Υπάρχουν δύο σημαντικά πλεονεκτήματα στο μοντέλο ελαχίστης προς τα πίσω ροής (MB – F)

1. Το (MB – F) δεν απαιτεί κόστος και προσπάθεια, διότι χρησιμοποιεί καθολικά δεδομένα σε ένα εργοστάσιο.

2. Λόγω της απλότητας του μοντέλου είναι εύκολο να υλοποιηθεί.

Ωστόσο προκειμένου να υλοποιηθεί το μοντέλο (MB – F), πρέπει να ικανοποιούνται οι παρακάτω προϋποθέσεις:

Επειδή τα δεδομένα που απαιτούνται για την υιοθέτηση του μοντέλου

(MB – F) είναι η καθορισμένη διεργασία και η απαίτηση των προϊόντων, ένα εργοστάσιο οφείλει να διατηρεί ακριβή δεδομένα γι' αυτά τα δύο πράγματα. Επίσης οποτεδήποτε γίνονται αλλαγές, πρέπει η χωροδιάταξη των μηχανών να αναδιοργανώνεται.

Ο κύριος στόχος αυτής της εργασίας είναι να ερευνηθεί η δυνατότητα του αλγορίθμου Ant Colony System (ACS), να εφαρμοστεί σε προβλήματα γραμμικής τοποθέτησης μηχανών. Ο αλγόριθμος ACS είναι αποτελεσματικός για τη λύση των προβλημάτων γραμμικής τοποθέτησης των μηχανών, βρίσκει γρήγορα τη βέλτιστη λύση εξοικονομώντας χρόνο. Παρατηρήθηκε ότι σε κάθε επανάληψη το κόστος για την παραγωγή των προϊόντων ελαχιστοποιείται. Η μηχανή που βρίσκει το ελάχιστο κόστος σημειώνεται σαν αποτέλεσμα και από αυτή αρχίζει η ακολουθία των υπολοίπων.

Ο τρόπος προσαρμογής και η αναφορά σε παραδοχές που σχετίζονται με το πρόβλημα της παρούσης εργασίας διαφοροποιεί τον κλασικό αλγόριθμο ACS με αποτέλεσμα να προκύπτει ένας αλγόριθμος με πρωτότυπα στοιχεία.

ΠΑΡΑΡΤΗΜΑ Ι

Αλγόριθμος Ant Colony System (DFACS) για την επίλυση προβλημάτων δρομολόγησης οχημάτων (VRP)(Κεφάλαιο 7^{ο})

Στη συνέχεια παρατίθεται ο πηγαίος κώδικας της εφαρμογής που παρουσιάστηκε παραπάνω, γραμμένος σε περιβάλλον Delphi

```
unit ACO;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, Db, DBTables, Grids, DBGrids, Math;

type

  TChance = record
    RouteNo, Chance: Integer;
  end;

  TAnt = class
    Capacity: Integer;
    public
      { Public declarations }
      CurNode : Integer;
      VisitedNodes : Array[0..100] of Integer;
      Constructor Create(aOwner : TComponent; aCapacity: Integer);
  end;
```

```
  TColony = class
    Ants: Array of TAnt;
    public
      { Public declarations }
      AntsNum : Integer;
      Constructor Create(aOwner : TComponent; aAntsNum, aAntCapacity:
Integer);
      Destructor Destroy;
  end;

  TACOFRM = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Table1: TTable;
    Label2: TLabel;
    Button3: TButton;
    DBGrid2: TDBGrid;
    DataSource2: TDataSource;
    Table2: TTable;
    Label1: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Table1NodeStart: TIntegerField;
    Table1NodeEnd: TIntegerField;
    Table1Cost: TIntegerField;
    Table1Phero: TIntegerField;
    Edit2: TEdit;
    Label4: TLabel;
    Edit3: TEdit;
    Label5: TLabel;
    Button1: TButton;
```

```
    Table1Routes: TIntegerField;
    Button2: TButton;
    Button4: TButton;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
    AntCap: Integer;
    TotalDemand, TotalUnloaded: Integer;
    AntsNum, ObjFunc : Integer;
    Colony : TColony;
    procedure InitValues;
    procedure RunSim;
    procedure ResetRoutes;
    procedure CalcObjFunc;
  public
    { Public declarations }
  end;

var
  ACOFRM: TACOFRM;

implementation
```

```
{$R *.DFM}

procedure TACOFRM.Button1Click(Sender: TObject);
begin
  Table1.ApplyUpdates;
  Table2.ApplyUpdates;
end;

procedure TACOFRM.Button2Click(Sender: TObject);
begin
  ResetRoutes;
end;

procedure TACOFRM.FormCreate(Sender: TObject);
begin
  Table1.Open;
  Table2.Open;
  ResetRoutes;
end;

procedure TACOFRM.ResetRoutes;
begin
  with Table1 do
  begin
    First;
    while not eof do begin
      Edit;
      FieldByName('Routes').AsInteger := 0;
      Post;
      next;
    end;
```

```
  end;
end;

procedure TACOFRM.InitValues;
Var FloatAnts: Double;
begin
  TotalDemand := 0;
  AntCap := StrToInt(Edit1.Text);
  with Table2 do begin
    first;
    While not EOF do begin
      TotalDemand := TotalDemand + FieldByName('DEMAND').AsInteger;
      Next;
    end;
  end;
  with Table1 do begin
    first;
    While not EOF do begin
      Edit;
      FieldByName('PHERO').AsInteger := StrToInt(Edit2.Text);
      Post;
      Next;
    end;
  end;
  FloatAnts := TotalDemand / AntCap;
  AntsNum := Round(FloatAnts);
  if AntsNum < FloatAnts then
    AntsNum := AntsNum + 1;
end;
```

```
procedure TACOFRM.Button3Click(Sender: TObject);
var
  i, PrevObjFunc: Integer;
begin
  Label9.Caption := EmptyStr;
  Label9.Repaint;
  Label7.Caption := EmptyStr;
  Label7.Repaint;
  InitValues;
  PrevObjFunc := maxint;
  for i:=1 to StrToInt(Edit3.Text) do begin
    Label7.Caption := IntToStr(i);
    Label7.Repaint;
    ResetRoutes;
    RunSim;
    CalcObjFunc;
    if PrevObjFunc < ObjFunc then
      ObjFunc := PrevObjFunc;
    PrevObjFunc := ObjFunc;
    ShowMessage(Format('Κύκλος: %d  Αντ/κή Συν/ση: %d', [i, ObjFunc]));
    Label9.Caption := IntToStr(ObjFunc);
    Label9.Repaint;
    table2.CancelUpdates;
  end;
end;

procedure TACOFRM.RunSim;

  procedure Calc(AntIndex: Integer);
  var
    i, j, z, low, CurTop, VisNod, ForbNod: integer;
```

```
  CanVisit: Boolean;
  Chances: Array[0..100] of TChance;
  Rnd, Unloaded, LastGoodNode, a, StartNode, EndNode: Integer;
  FoundDest: Boolean;
  ForbNodes: Array[0..100] of Integer;
begin
  VisNod := -1;
  ForbNod := -1;
  With Colony.Ants[AntIndex] do
  begin
    if Capacity = 0 then
      CurNode := 1
    else while (Capacity>0) do
    begin
      j := 0;
      CurTop := 0;
      FoundDest := False;
      Table1.First;
      While not Table1.EOF do
      begin
        if (Table1.FieldByName('NODESTART').AsInteger = CurNode) then
        begin
          Table2.Locate('ID', Table1.FieldByName('NODEEND').AsInteger,
[]);
          CanVisit := true;
          if CanVisit then begin
            for i:=0 to VisNod do
              if Table1.FieldByName('NODEEND').AsInteger =
VisitedNodes[i] then
                CanVisit := False;
            for i:=0 to ForbNod do
```

```
            if Table1.FieldByName('NODEEND').AsInteger = ForbNodes[i]
then
              CanVisit := False;
          end;
          if CanVisit then begin
            FoundDest := true;
            Chances[j].RouteNo := Table1.RecNo;
            Chances[j].Chance := CurTop +
Table1.FieldByName('Phero').AsInteger;
            CurTop := Chances[j].Chance;
            j := j + 1;
          end;
        end;
        Table1.Next;
      end;
      if FoundDest then begin
        Randomize;
        Rnd := Round(Random(CurTop));
        low := 0;
        For z:=0 to j-1 do
        begin
          if (low<=Rnd) and (Rnd<=Chances[z].Chance) then
          begin
            Table1.RecNo := Chances[z].RouteNo;
            CurNode := Table1.FieldByName('NodeEnd').AsInteger;
            Table2.Locate('ID', Table1.FieldByName('NODEEND').AsInteger,
[]);
            unloaded := min(Capacity,
Table2.FieldByName('Demand').AsInteger);
            Table1.Edit;
              Table1.FieldByName('Phero').AsInteger :=
```

```
            Table1.FieldByName('Phero').AsInteger +
Round(100000/(Table1.FieldByName('Cost').AsInteger));
          TotalUnloaded := TotalUnloaded + unloaded;
          if unloaded > 0 then begin
           LastGoodNode := VisNod + 1;
           Table2.Edit;
           if Table2.FieldByName('Demand').AsInteger >0 then
            Table2.FieldByName('Demand').AsInteger :=
             Table2.FieldByName('Demand').AsInteger - unloaded;
           Table1.FieldByName('Routes').AsInteger :=
            Table1.FieldByName('Routes').AsInteger + 1;
           Table2.Post;
          end;
          Table1.Post;
          Capacity := Capacity - unloaded;
          VisNod := VisNod + 1;
          VisitedNodes[VisNod] :=
Table1.FieldByName('NODEEND').AsInteger;
         end;
         low := Chances[z].Chance;
        end;
       end
       else begin
        ForbNod := ForbNod + 1;
        ForbNodes[ForbNod] := CurNode;
        VisNod := VisNod - 1;
        if VisNod > -1 then begin
         CurNode := VisitedNodes[VisNod];
         Table1.Locate('NodeStart;NodeEnd',
            VarArrayOf([VisitedNodes[VisNod], VisitedNodes[VisNod+1]]),
[]);
```

```
          Table1.Edit;
          Table1.FieldByName('Phero').AsInteger :=
            Table1.FieldByName('Phero').AsInteger -
Round(100000/(Table1.FieldByName('Cost').AsInteger));
          Table1.Post;
        end else begin
          VisNod := -1;
          CurNode := 1;
          if TotalUnloaded = TotalDemand then
            Capacity := 0;
        end;
      end;
    end;
  end;

var
  i: integer;
begin
  TotalUnloaded := 0;
  Colony := TColony.Create(Self, AntsNum, AntCap);
  with Colony do
  begin
    for i:=0 to AntsNum-1 do
    begin
      Calc(i);
    end;
  end;
end;

procedure TACOFRM.CalcObjFunc;
```

```
begin
  ObjFunc := 0;
  With Table1 do
  begin
    First;
    while not eof do
    begin
      ObjFunc :=
        ObjFunc + (FieldByName('Routes').AsInteger *
FieldByName('Cost').AsInteger);
      Next;
    end;
  end;
end;

{ TAnt }

constructor TAnt.Create(aOwner: TComponent; aCapacity: Integer);
begin
  Capacity := aCapacity;
  CurNode := 1;
end;

{ TColony }

constructor TColony.Create(aOwner: TComponent; aAntsNum,
aAntCapacity: Integer);
var
  i: Integer;
  Ant: TAnt;
begin
```

```
  SetLength(Self.Ants, aAntsNum);
  For i:=0 to aAntsNum-1 do
  begin
    Ant := TAnt.Create(aOwner, aAntCapacity);
    Self.Ants[i] := Ant;
    Ant := nil;
  end;
  Self.AntsNum := aAntsNum;
end;

destructor TColony.Destroy;
var i: Integer;
begin
  for i:=0 to AntsNum-1 do
    Ants[i].Free;
end;

end.
```

ΠΑΡΑΡΤΗΜΑ ΙΙ

Ψευδοκώδικας του Αλγορίθμου MAX – MIN Ant System (Κεφάλαιο 9^{ο})

Start

// Initialize

for i:1 to n-1 (έστω ότι έχουμε n σταθμούς)

for j:1 to end (χρησιμοποιήσαμε την ένδειξη “end” για να δείξουμε ότι δεν απαιτείται όλες οι μηχανές να έχουν τον ίδιο αριθμό στάθμεων ισχύος. Ο χωρισμός μπορεί να γίνει όπως είπαμε στην αρχή με διαφόρους τρόπους, και αρκεί να κρατάμε σε μία λίστα (πίνακα) τις στάθμες για κάθε μηχανή – σταθμό)

set $\tau_{ij}(0) = \tau_0$ (πολύ μεγάλη τιμή)

end for

end for

for i:1 to n-1

for j:1 to end

$n_{ij}(t) = P_{ij} / F_i(P_{ij})$

(ή το πιο απλοϊκό $n_{ij}(t) = 1 / F_i(P_{ij})$)

end for

end for

for k : 1 to K

set every ant to one machine (αρχικοποίηση της Tabu list για κάθε μυρμήγκι – Tabu list είναι απλά ένα διάνυσμα που καταγράφει τις μηχανές από τις οποίες πέρασε το κάθε μυρμήγκι και τη στάθμη που επέλεξε για κάθε μία από αυτές)

end for

Initialize minimum Cost, $\mathrm{Cost}_{\mathrm{best}}$ (σε μία μεγάλη τιμή)

//Main Loop

for t : 1 to maximum iterations

for k=1 to K

for i=1 to n-1

select Power level (using the transition rule)

Put the power level in Tabu list: $\mathrm{T}^{\mathrm{k}}(\mathrm{t})$

End for

Compute $\mathrm{Cost}, \mathrm{Cost}^{\mathrm{k}}(\mathrm{t})$

If $\mathrm{Cost}^{\mathrm{k}}(\mathrm{t}) < \mathrm{Cost}_{\mathrm{best}}$ set $\mathrm{Cost}_{\mathrm{best}} = \mathrm{Cost}^{\mathrm{k}}(\mathrm{t})$ and keep edges of best configuration $\mathrm{T}_{\mathrm{best}} = \mathrm{T}^{\mathrm{k}}(\mathrm{t})$

Re evaluate $\tau_{\max}, \tau_{\min}$

End if

End for

For i=1 to n-1

For j=1 to m

Update pheromone trails by applying the pheromone update rule using mainly $\left(\mathrm{Cost}_{\mathrm{iter}}(\mathrm{t})\right)$ (or optional) $\left(\mathrm{Cost}_{\mathrm{best}}(\mathrm{t})\right)$ – Η επιλογή του $\left(\mathrm{Cost}_{\mathrm{best}}(\mathrm{t})\right)$ θα πρέπει να είναι περιορισμένη σε σχέση με τη χρησιμοποίηση της καλύτερης ανά επανάληψη λύση)

End for

End for

End for

If T_{best} feasible
Print $Cost_{best}$ T_{best}
Else if
Prompt for re-initialization of the algorithm
End if

Stop

Παρατήρηση:
Στον παραπάνω ψευδοκώδικα δεν κάνουμε έλεγχο αν η λύση είναι δυνατή (δηλαδή να μη παραβιάζονται τα όρια ισχύος για τη μηχανή αναφοράς) παρά μόνο στο τέλος. Υποθέτουμε ότι το κόστος της παραβίασης θα είναι τέτοιο που μη αποδεκτές λύσεις δεν θα συγκαταλέγονται στις «καλές» λύσεις.
Μπορεί να προστεθεί ένα κομμάτι που απλά να κάνει έλεγχο, να αγνοεί τη λύση που βρέθηκε να μην ανανεώνει τη φερομόνη και να περνάει στην επόμενη επανάληψη.

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Ψευδοκώδικας του Αλγορίθμου ACS
για τη λύση του προβλήματος γραμμικής τοποθέτησης μηχανών

Start

// Initialize

for i,j:1 to m
    set $\tau_{ij}(1) = \tau_0$
end for

for i,j:1 to m ($i \neq j$ αλλιώς θα έχουμε διαίρεση με το μηδέν)
    $n_{ij} = 1/d_{ij} \quad (n_{ij} = 0)$
end for

for k:1 to m
    set every ant to one machine (αρχικοποίηση της Tabu list για κάθε μυρμήγκι – Tabu list είναι απλά ένα διάνυσμα που καταγράφει τις μηχανές από τις οποίες πέρασε το κάθε μυρμήγκι)
end for
Initialize minimum Cost, $Cost_{best}$ (σε μια μεγάλη τιμή)

//Main Loop

for t : 1 to maximum iterations

    for k=1 to m

        reinitialize Tabu list (erase the list that has been created during the previous iteration)
        for i=1 to m-1
            select next machine (using the transition rule)
            Put next machine in Tabu list: $T^k(t)i \leq m-1$
        End for

Compute Cost, $Cost^k(t)$

If $Cost^k(t) < Cost_{best}$ set $Cost_{best} = Cost^k(t)$ and keep edges of best layout: $T_{best} = T^k(t)$

End for

For i,j=1 to m

Update pheromone trails by applying the pheromone update rule

End for

End for

End for

Print $Cost_{best}$ T_{best}

Stop

ΒΙΒΛΙΟΓΡΑΦΙΑ

AARTS, E., KORST, J., (1989). Simulated Annealing and Boltzmann Machines, John Wiley and Sons.

ALEXANDRIS, N,. FOUNTAS, C., VLACHOS, A., (2005). The ant Colony system: optimization for the logistics of marine cargo in the Aegean, *Journal of Statistics and Management Systems*, 8, (1), 1-11.

ALONSO, S., CORDON, O., FERNADEZ DE VIANA, I., HERRERA, F., (2004). Integrating Evolutionary Computation Compouters in Ant Colony Optimization Evolutionary Algorithms : An Experimental study. *Recent Developments in Biologically Inspired Computing*, L. Nunes de Castro, F. J. Von Zuben (Eds), Idea Group Publishing, 148-180

ANEKE, N.A.G., AND CARRIE, A.S., (1986). A Design for the Layout of Multi-product Flow lines. *International Journal of production Research*, (24), No: 3, 471-481.

BECKERS, R., DENEUBOURG, J. L., GOSS, S., (1992). Trai1s and U-turns in the Selection of a Path by the Ant, *Lasius niger*, J. theor. Biol., 159, 397-415.

BIRATTARI, M., DI CARO, G., AND DORIGO, M., (2002). Toward the Formal Foundation of Ant Programming. Proceeding of the Third International Workshop on Ant Algorithms (Ants' 2002), edité par M. Dorigo, G. Di Caro et al., tome 2463 de *Lecture Notes in Computer Sciènce*, 188-200. Springer Verlag, Brussels, Belgium.

BOLONDI, M., AND BONDANZA, M., (1993). Parallelizzazione di un algoritmo per la risoluzione del problema del couresso viaggiatore Master's thesis, Dipartimento de Elettronica, Politecuico di Milano, Italy.

BONADEAU, E., THERAULAZ, G., DENEUBOURG, J. L., ARON, S., AND CAMAZINE, S., (1997). Self-organization in Social Insects, Trends in *Ecol. Evol*, 12 188-193.

BONABEAU, E., DORIGO, M., THERAULAZ, G., (1999). Swann Intelligence,

From Natural to Artificial Systems, Oxford University Press.

BRUISMA, O.H., (1979). An Analysis of Building Behaviour of the Termite, *Macrotermes Subhyalinus* (Rambur). Thesis, The Netherlands: Landbouwhoge School, Wageningen.

BULLNHEIMER, B., HARTL, R. F., STRAUSS, C., (1999). A new rank-based version of the Ant System: A computational study, *Central European Journal for Operations Research and Economics*, 7, 1, 25-38.

CAMAZINE, S., DENENBOURG, J. L., FRANKS, N. R., SHYED, J., THERAULAZ, G. AND BOUNABEAU, E., (Eds), (2001). Self-Organization in Biological Systems, Princeton University Press.

CHEBBO, A. M., IRVING, M. R., ( ). Combined active and reactive dispatch, Proc. *IEE*, *Pt. C.*, 4, 393-405

CHERNY, V., (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, 45, 41-51.

CONTAXIS, G. HATZIARGYRIOU, N., ( ). Power System operation and control (in Greek), National Technical University of Athens, Department of Electrical and computer Engineering, Electric Power Division, 238-243.

COOPER, L., (1965). Location – Allocation Problems, *Opns. Res*., 11, 331-343.

CORDON, O., HERRERA, F., STÜTZLE, T., (2002). Ant Colony Optimization, Part Four, 190, Parpineli, Lopes and Freitas.

CORNE, D., DORIGO, M., AND GLOVER, F., (Editors), (1999). New Ideas in Optimization, 63-77, McGraw-Hill.

DAMMEYER, F., VOB, S., (1993). Dynamic tabu list management using the reverse elimination method, *Annals of Operatins Research*, 41. 31-46.

DARCHEN, R., (1959). Les Techiques de la Construction chez Apis

Mellifica, Ph.D., dissertation, Université de Paris.

DENEUBOURG J. L., ARON, S., GOSS, S., AND PASTEELS, J. M., (1990). The Self-Organizing exploratory pattern of the Argentine ant, Journal of lnsect Behavior, 3, 159-168.

DORIGO, M., (1992). Ottimizzazione, Apprendirnento Automatico, ed Algoritmi Basati su Metafora Naturale. Ph.D. Dissertation, Politecnico di Milano, Italy.

DORIGO M., AND GAMBARDELLA, L. M., (1996). A study of some Properties of Ant-Q. In Proceedings Fourth International Conference on Parallel Problem, Solving From Natura, PPSN IV.

DORIGO, M., MANIEZZO, V., COLORNI, A., (1996). The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. System Man Cybern, B26, 29-41.

DORIGO M., AND GAMBARDELLA, L. M., (1997). Ant Colony System: A cooperative learning Approach to the Travelling Salesman Problem, *IEEE Trans. Evo1. Comp.* 1, 53-66.

DORIGO M., DI CARO, G. AND GAMBARDELLA, L. M., (1999). Ant Algorithms for discrete Optimization. *Artificial Life*, 5, 137-172.

DORIGO M., BONABEAU, E., AND THERAULAZ G., (2000). Ant algorithms and stigmergy, *Future Generation Computer Systems*, 16 (8), 851-871.

DORIGO M., AND STÜTZLE, T., (2003). *Handbook of Metaheuristics*, tome 57, de International Series in Operations research and managent Science, chapitre the Ant Colony Optimization Metaheurstics: Algorithms, Applications and Advances. Kluwer Academic Publishers, Boston Hardbound.

DORIGO M., AND STÜTZLE, T., (Editors), (2004). Ant Colony Optimization, MIT Press. Eglese, R. M., (1990). Simulated annealing: A tool for Operatinal Research, European *Journal of Operational Research*, 46, 271-281.

EUROPEAN COMMISSION, (2000), M. C. M., Personal Communication.

FENEKOS, G., (2001). Ant Colony Optimization, Master thesis, National Technical University of Athens, Department of Mechanical Engneering.

FOUNDAS, C., AND VLACHOS, A., (2005). Ant Colonies Optimization (ACO) for the solution of the Vehicle Routing Problem (VRP). *Journal of Information and Optimization Sciences*, 26 (1), 135-142.

FOUNDAS, C., AND VLACHOS, A., (2005). Optimal Solution of linear machine layout Problem using Ant Colony System, *Wseas Transactions on Information Science and applications*. Issue 6, 2, 652-661.

FOUNDAS, E., AND VLACHOS, A., (2005). Same application of difference equations – Ant Colony Otpimization (ACO), accepted for publication in *Journal of Interdisciplinary Mathematics*.

FOUNDAS, E., AND VLACHOS, A., (2005). New approaches to evaporation in Ant Colony Optimization algorithms, accepted for publication in *Journal of Interdisciplinary Mathematics*.

GAMBARDELLA, L. M., AND DORIGO M., (1995). Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In Proceedings Twelfth International Conference on Machine Learning, ML-95, 252-260, Paolo Alto, CA: Morgan Kaufmann.

GAREY, M. R., AND JOHNSON, D.S., (1979). Computers and Intractability (Freeman, San Francisco).

GLOVER, F., (1986). Future paths for integer programming and links to artificial intelligence, Computers and Operation Research, 13, 533-549.

GLOVER, F., LAGUNA, M., (1993). Tabu Search, chapter in *Modern Heuristic Techniques for Combinatorial Problems*, C. Reeves, ed., Blackwell Sientific Puublishing, 71-140.

GLOVER, F., (1993). A user's guide to tabu search, *Annals of Operations Research*, 41, 3-28.

GLOVER, F., (1994). Genetic Algorithms and Scatter Search: Unsuspected Potentials, *Statistics and Computing*, 4, 131-140.

GOLDBERG, D., (1989). Genetic Algorithms in Search, *Optimization and Machine Learning*, Reading, MA: Addison – Wesley.

GOSS, S., ARON, S., DENEUBOURG J. L., PASTEELS, J. M., (1989). Self-Organized shortcuts in the Argentine Ant. *Naturwissenschaften* 76, 579-581.

GRASSÉ, P. P., (1959). La Reconstruction dun id et les Coordinations Futer-Intividuelles chez, *B ellicositermes Natalensis et Cubitermes sp*. La theorie de la stigmergie: Essai d' interprétation du Comportement des Termites Coustructeurs. *Insect-soc.*, 6, 41-80.

GUAN, X., LUH, P. B., ZHANG, L., (1995). Non linear approximation method in Lagrangian relaxation based algorithms for hydrothermal scheduling, *IEEE Trans. Power Systems*, 10 (2), 772-778.

HAREL, D., (1987). Algorithms, The Spirit of computing, Addison-Wesley.

HEINRICH, B., (1981). The Regulation of Temperature in the Honey Bee Swarm. *Sci Am*, 146-160.

HERTZ, A., KOBER, D., (2000). A framework for the description of evolutionary algorithms, *European Journal of Operational Research*, 126, 1-12.

HOLLAND, J. H., (1975). Adaption in Natural and Artificial Systems, The University of Michigan Press, Ann Arbor, MI.

HOLLDOBLER, B., AND WILSON, E. O., (1990). The Ants. Cambridge, MA: Harvard University Press.

JEANNE, R. L., (1986). The Evolution of the Organization of Work in Social Insects, *Monit. Zool. Ital.*, 20, 119-133.

JEANNE, R. L., (1996). Regulation of Nest Construction Behaviour in Polybia occidentalis, *Anim. Behav*., 52, 173-488.
IREDI, S., MERKLE, D., AND MIDDENDORF, M., (2001). Bi-Criterion Optimization with Multi Colony Ant Algorithms, presented at Evolutionary Multi-Criterion Optimization, First International Conference (EMO '01), Zurich.

KIRIKPATRICK, S., GELLAT, L., VECCHI, M., (1983). Optimization by Simulated annealing, *Science*, 220, 671-680.

LAWLER, E. L., LENESTRA, J. K., RINNOOY-KAN, A. H. G., AND SHMOYS D., (1985), eds. *The travelling Salesman Problem*. New York, NY: Wiley and Sons.

LEE, C. E. C., (1991). An Integrated Methodology for the Analysis and Disign of Cellular Flexible Assembly Systems, Ph. D. Dissertation, Purdue University, West Lafayette, IN, USA.
LUSCHER, M., (1961). Air-Conditioned Termite Nests. *Sci.Am*. 205, 138-145.

MAETERLINCK, M., (1927). *The Life of the White Ant*. London: George Allen and Unwin.

MANTAWY, A. H., ABDEL-MAYID, Y. L., SELIM, S. Z., (1998). Unit Commitment by tabu Search, IEE Proceeding *Generation, Transmission and Distribution*, 145 (1), 55-64.

MCCULLOCH, W. S., PITTS, W., (1943). A Logical calculus of the Ideas Immanent in Nervons Activity, *Bulletin of Mathematical Biophysics*, 5, 155-133.

METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A., TELLER, E., (1953). Equation of State Calculations by fast computing machines, *Journal of chemical physics*, 21, 1087-1092.

MONTGOMERY, J., AND RANDALL, M., (2002). Anti-pheromone as a Tool

for Better Exploration of Search Space. Ant Algorithms, KNCS 2463, Seiten 100-110. Berlin, Spinger Verlag.

MOSHE, E. C., MEHREZ, A., MARKOVISH, G., (2002). *Computers and operations research*, 29, 429-470.

PANTA, L., (2002). Modelling Transportation Problem Using Concepts of Swarm Intelligence and Soft Computing. Ph.D, Falls Church Virginia.

PRICE, K., (1996). Differential Evolution: a fast and simple numerical Optimizer, in M. Smith M., *Processing Society*, IEEE Press, New York, NY, 524-527.

REEVES, C., (1993). Modern Heuristic Techniques for Combinatorial Problems. Mc Graw-Hill.

REINELT, G., (1991). TSPLIB-A travelling salesman problem library. *ORSA Journal on Computing*, 3, 376-384.

ROCHAT, Y., TAILLARD, E. D., (1995). Probabilistic Diversification and Intensification in Routing, *Journal of Heuristics*, 1, 147-167.

SABRACOS, E. (2000). Exploring Operational Problems of the Goods Supply chain in Greek Islands: Towards a Reengineering of the System Repositioning Logistics, 16th International Logistics Congress, Versailles.

SCHOONDERWOERD, R., HOLLAND, O., BRUTEN, J., AND ROTHKRAUTZ, L., (1996). Ant-Based Load Balancing in Telecommunications Networks, *Adapt. Behav*. 5, 169-207.

STÜTZLE, T., AND HOOS, H., (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16, 889-914.

VLACHOS, A., (2005). Ant Colony system solving Capacitated location –

allocation problems on a line, accepted for publication in *Journal of Information and Optimization Science*.

VLACHOS, A., AND MOUE, A., (2005). Max-Min Ant System Applied in Economic Load Dispatch, WSEAS TRANSACTIONS ON COMPUTERS, Issue 7, 4, 711-717.

THERAULAR, G., BONABEAU, E., (1999). A brief history of stigmergy, *Artificial Life*, 97-116.

WILSON, E. O., (1975). *Sociobiology* Cambridge, MA: Harvard University Press.

WONG, K. P., FUNG, C. C., (1993). Simulated anneling based economic dispatch algorithm, *IEE Proc. Pt. C*, 509-515.

WON, YOU-DONG, (1997). A Linear Programming Approach to Linear Machine Layout Problem, *The Journal of the Industrial Mathematics Society*, 47, Part 2.