Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Αλγόριθμοι Τεχνητής Ευφυΐας Σμήνους και Εφαρμογές σε Μη-ντετερμινιστικά Πολυωνυμικά Προβλήματα

Φούντας Χρυσόστομος

Διδακτορική Διατριβή

Πειραιάς 2008

Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Διατριβή

Για την απόκτηση Διδακτορικού Διπλώματος του Τμήματος Πληροφορικής

Χρυσόστομου Ε.Φούντα

¨Αλγόριθμοι Τεχνητής Ευφυΐας Σμήνους και Εφαρμογές σε Μη-ντετερμινιστικά Πολυωνυμικά Προβλήματα¨

Συμβουλευτική Επιτροπή :

Επιβλέπων:

Νικόλαος Αλεξανδρής
Καθηγητής Πανεπιστημίου Πειραιώς

Μέλη:

Γεώργιος Τσιχριντζής
Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Γρηγόριος Χονδροκούκης
Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Εξεταστική Επιτροπή:

Νικόλαος Αλεξανδρής
Καθηγητής Πανεπιστημίου Πειραιώς

Νικόλαος Μπλέσιος
Καθηγητής Πανεπιστημίου Πειραιώς

Αριστείδης Σαπουνάκης
Καθηγητής Πανεπιστημίου Πειραιώς

Αγγελική Βουδούρη
Καθηγήτρια Πανεπιστημίου Αθηνών

Γεώργιος Τσιχριντζής
Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Γρηγόριος Χονδροκούκης
Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Δημήτριος Αποστόλου
Λέκτορας Πανεπιστημίου Πειραιώς

…στη ζωή που έρχεται.

Ευχαριστίες

Η παρούσα διδακτορική διατριβή ολοκληρώθηκε με την αμέριστη συμπαράσταση, ηθική και επιστημονική των μελών της τριμελούς συμβουλευτικής επιτροπής. Θα ήθελα να ευχαριστήσω:

Τον επιβλέποντα καθηγητή κ. Νικόλαο Αλεξανδρή ο οποίος με την εμπειρία, τις γνώσεις , τις υποδείξεις και τις εύστοχες παρατηρήσεις του συμπαραστάθηκε απόλυτα στην ερευνητική προσπάθεια μου και με βοήθησε τα μέγιστα στην ολοκλήρωση της εργασίας αυτής.

Τον αναπληρωτή καθηγητή κ. Γεώργιο Τσιχριντζή, μέλος της τριμελούς συμβουλευτικής επιτροπής, για τη σημαντική συμβολή του στην ολοκλήρωση της ερευνητικής προσπάθειας μου.

Τον αναπληρωτή καθηγητή κ. Γρηγόριο Χονδροκούκη, μέλος της τριμελούς συμβουλευτικής επιτροπής, για τις ενδιαφέρουσες συζητήσεις και παρατηρήσεις που με βοήθησαν στο έργο μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την άνευ όρων υποστήριξη και απεριόριστη υπομονή τους.

Περίληψη

Η παρούσα διδακτορική διατριβή ασχολείται με τα Μη ντετερμινιστικά Πολυωνυμικά Προβλήματα και τις λύσεις που προτείνονται για αυτά από τους Αλγορίθμους Ευφυΐας Σμήνους. Προβλήματα με πλήθος εφαρμογών μπορούν να αναχθούν σε προβλήματα της κλάσης Non deterministic Polynomial και οι τροποποιήσεις τους με σκοπό να καλύψουν μεγαλύτερο εύρος εφαρμογών δείχνουν το ενδιαφέρον που έχουν προκαλέσει στη ερευνητική κοινότητα καθώς και την αντίστοιχη προσπάθεια που έχει γίνει για την επίλυση τους. Τα προβλήματα που ερευνήθηκαν στην παρούσα διατριβή όπως, το Πρόβλημα Περιοδεύοντος Πωλητή, το Πρόβλημα Δρομολόγησης Οχημάτων, το Πρόβλημα Τετραγωνικής Ανάθεσης υπό τη μορφή του Προβλήματος Γραμμικής Τοποθέτησης Μηχανών προσεγγίζονται με παλαιότερες μεθόδους, όπως της Προσομοιωμένης Ανόπτησης, αλλά και με νέους αλγορίθμους σμήνους ευφυΐας μυρμηγκιών. Γίνεται εκτενής έρευνα της διεθνούς αρθρογραφίας και βιβλιογραφίας με σκοπό την επιλογή των βέλτιστων αλγορίθμων και τη κατανόηση της συμπεριφοράς των αλγορίθμων σε σχέση με τα υπό έρευνα προβλήματα. Τέλος προτείνεται και ελέγχεται η απόδοση ενός νέου αλγορίθμου σμήνους ευφυΐας μυρμηγκιών.

Περιεχόμενα

Κεφάλαιο 3

3. Η Κοινωνική Συμπεριφορά των Εντόμων και η Βιολογική Προέλευση των Ιδεών

Κεφάλαιο 4

4. Η οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών

Κεφάλαιο 5

5. Εναλλακτικές Μεθοδολογίες

Κεφάλαιο 6

6. Βελτιστοποίηση Θαλάσσιων Μεταφορών με τον Αλγόριθμο Προσομοιωμένης Ανόπτυσης

Κεφάλαιο 7

7. Εφαρμογές του Αλγορίθμου Ant Colony System

Κεφάλαιο 8

8. Μια Νέα Προσέγγιση του Αλγορίθμου Ant Colony System και Εφαρμογή για τη Βελτιστοποίηση του Προβλήματος του Περιοδεύοντος Πωλητή

Κεφάλαιο 9

9. Συμπεράσματα

Κεφάλαιο 1

Εισαγωγή

1.1 Το έναυσμα για την έρευνα

Μεγάλο πλήθος προβλημάτων πρακτικής και θεωρητικής φύσης σχετίζεται με την επιλογή της 'άριστης' διαμόρφωσης ή την βέλτιστη επιλογή παραγόντων για την επίτευξη ενός στόχου.

Ένα Πρόβλημα Συνδυαστικής Βελτιστοποίησης είναι ένα πρόβλημα σε ένα πεπερασμένο σύνολο επιλογών που πρέπει είτε να μεγιστοποιηθεί, είτε να ελαχιστοποιηθεί και αναλυτικά η μελέτη τέτοιων προβλημάτων παρουσιάζεται στο κεφάλαιο 2. Για αρκετά χρόνια τα Προβλήματα Συνδυαστικής Βελτιστοποίησης απασχολούν τους ερευνητές καθώς οι κομψές και σχετικά απλές μαθηματικές τους εκφράσεις αρχικά παραπλανούν, καθώς μετά την συνδυαστική έκρηξη του χώρου αναζήτησης αποκαλύπτεται η πραγματική φύση του προβλήματος.

Τα διακριτά προβλήματα βελτιστοποίησης έχουν ένα μετρήσιμο πεπερασμένο σύνολο εφικτών λύσεων, καθώς και δείκτη για την μέτρηση τους, την αντικειμενική συνάρτηση κάθε λύσης. Η διακριτή βελτιστοποίηση βρίσκει την καλύτερη λύση βάση της αντικειμενικής συνάρτησης ανάμεσα στις εφικτές λύσεις. Πολλά τέτοιου είδους προβλήματα ανήκουν στην κλάση προβλημάτων NP-Hard, πράγμα που σημαίνει ότι δεν υπάρχει αλγόριθμος που να μπορεί να δώσει την απόλυτα βέλτιστη λύση σε πολυωνυμικό χρόνο.

Αλγόριθμος είναι ένας συνδυασμός ενεργειών που λύνει κάθε στιγμιότυπο ενός δοσμένου προβλήματος και μπορεί να προγραμματιστεί σε κάποια υπολογιστική μηχανή. Όταν δίνεται ένα σύνολο πληροφοριών σαν εισαγόμενα δεδομένα και ζητείται ένα νέο είδος εξαγόμενων δεδομένων, έχουμε το στιγμιότυπο ενός προβλήματος.

Υπάρχουν αρκετοί αλγόριθμοι που προσπαθούν να λύσουν προβλήματα συνδυαστικής βελτιστοποίησης. Πολλά πραγματικά προβλήματα μοντελοποιήθηκαν σαν προβλήματα συνδυαστικής βελτιστοποίησης και οδήγησαν τις τελευταίες τέσσερις δεκαετίες σε πλήθος αλγοριθμικών μεθόδων. Με δεδομένο ότι η εύρεση της απόλυτα βέλτιστης λύσης σε πολυωνυμικό χρόνο είναι ασύμφορη υπολογιστικά, γεννιέται το ερώτημα πόσο γρήγορα και πόσο ποιοτική λύση μπορούμε να προσεγγίσουμε. Οι προσεγγιστικοί αλγόριθμοι είναι η λύση για τα NP-Hard υπολογιστικά προβλήματα.

Η ευριστική αναζήτηση (heuristic search), που πολλές φορές αναφέρεται και σαν προσεγγιστικός αλγόριθμος, είναι η χρήση ενός πρακτικού κανόνα, μιας απλοποίησης, που έχει σαν σκοπό να περιορίσει το χώρο αναζήτησης σε ιδιαίτερα δύσκολα προβλήματα. Οι ευριστικές μέθοδοι σκοπεύουν στην υπολογιστική απόδοση και την απλότητα της ιδέας με πιθανό κόστος την ορθότητα και/ή ακρίβεια των λύσεων. Δηλαδή απεμπολώντας την εγγύηση της απόλυτα βέλτιστης λύσης, κερδίζουμε λύσεις που είναι πάρα πολύ κοντά στην θεωρητικά άριστη με τεράστιο κέρδος σε υπολογιστικό χρόνο.

Ο όρος μεταευριστική αναζήτηση ορίζεται ως μια κυρίαρχη στρατηγική η οποία κατευθύνει και τροποποιεί άλλες ευριστικές μεθόδους για να παράγει λύσεις πέραν αυτόν που κανονικά παράγονται κατά την αναζήτηση τοπικού βέλτιστου. Από τις πρώτες μεταευριστικές μεθόδους είναι οι Tabu Search, Προσομοιωμένη Ανόπτηση (simulated annealing) και οι γενετικοί αλγόριθμοι (genetic algorithms) οι οποίες αναλύονται σε επόμενο κεφάλαιο.

Η Ευφυΐα Σμήνους (swarm intelligence) είναι τεχνητή ευφυΐα που βασίζεται στη συλλογική συμπεριφορά αποκεντρωμένων, αυτο-οργανομένων συστημάτων. Ο πληθυσμός των ατόμων του συστήματος αποτελείται από άτομα που είτε είναι όμοια είτε διαφέρουν ελάχιστα μεταξύ τους αλλά είναι σχετικά απλά σαν μονάδες. Αυτό ο πληθυσμός αλληλεπιδρά δίχως να υπάρχει ένα γενικό πλαίσιο κανόνων ή συνειδητή συμπεριφορά των ατόμων και το σύνολο των αλληλεπιδράσεων εκφράζει τη συλλογική ευφυΐα του συστήματος. Αυτό είναι απόρροια της τοπικής αλληλεπίδρασης ατόμων που γειτνιάζουν στο χώρο και της συμπεριφοράς τους που ορίζεται από απλούς κανόνες. Συχνά αυτή η συμπεριφορά του κάθε ατόμου περιγράφεται με στοχαστικούς όρους που βασίζονται στη τοπική αντίληψη του ατόμου για το περιβάλλον του.

Η Ευφυΐα Σμήνους περιλαμβάνει φυσικά και τεχνητά συστήματα που αποτελούνται από πληθυσμό ατόμων που συντονίζονται με αποκεντρωμένο έλεγχο και αυτο-οργάνωση. Παραδείγματα τέτοιων συστημάτων φύση είναι οι αποικίες μυρμηγκιών και τεχνητές εφαρμογές με τέτοιες συμπεριφορές είναι πολύ-ρομποτικά συστήματα και λογισμικό με σκοπό τη βελτιστοποίηση.

Σχετικά πρόσφατες αλγοριθμικές μέθοδοι με σκοπό την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης εμπνέονται από τη φύση. Τέτοιες μέθοδοι περιγράφονται στην οικογένεια μεταευριστικών αλγορίθμων μυρμηγκιών (Ant Colony Optimization) που βασίζονται σε μελέτη της συλλογικής συμπεριφοράς των κοινωνικών αυτών εντόμων.

Τα μυρμήγκια είναι κοινωνικά έντομα με σύνθετη συλλογική συμπεριφορά η οποία είναι εμφανής όταν παρατηρηθούν συστηματικά. Ένα παιδί που παίζει σίγουρα θα δει αυτά τα μικρά έντομα στο χώρο και πιθανώς να παρατηρήσει ότι κινούνται σε μια γραμμή που αν την ακολουθήσει επιστρέφει στη φωλιά. Είναι μάλλον δύσκολο για το παιδί να σκεφθεί ότι τα μυρμήγκια δημιουργούν διαδρομές ελάχιστου μήκους από την πηγή τροφής στη φωλιά. Ακόμα πιο δύσκολο φαίνεται ότι ήταν οι ενήλικες να κατανοήσουν πώς αναδύεται αυτή η συλλογική συμπεριφορά της αποικίας μυρμηγκιών. Οι πρώτες επιστημονικές απόπειρες για την εξήγηση του φαινομένου εμφανίστηκαν στις αρχές της δεκαετίας του 90 όπου το βιολογικό μοντέλο συμπεριφοράς έδειχνε τη φερομόνη, μια χημική ουσία που εκκρίνουν τα μυρμήγκια, ως ρυθμιστικό παράγοντα στο σχηματισμό των διαδρομών. Από εκεί το βήμα στο τεχνητό μοντέλο αποικίας μυρμηγκιών ήταν θέμα χρόνου, καθώς ήταν προφανής η πιθανολογούμενη χρησιμότητα τους σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης.

Η βασική ιδέα είναι ότι οι αρχές της αυτο-οργάνωσης που επιτρέπουν την άριστα συντονισμένη ενέργεια χιλιάδων μυρμηγκιών μπορούν να χρησιμοποιηθούν για να συντονιστεί ένας πληθυσμός τεχνητών μυρμηγκιών που σκοπό έχουν την λύση υπολογιστικών προβλημάτων. Το κεφάλαιο 3 δείχνει το σχηματισμό των πρώτων ιδεών για τη λειτουργία αυτών των συστημάτων ευφυΐας σμήνους και την μετάβαση από το βιολογικό στο τεχνητό μοντέλο.

Το κεφάλαιο 4 παρουσιάζει την οικογένεια μεταευριστικών αλγορίθμων βελτιστοποίησης μυρμηγκιών και τις διαφορετικές ιδέες που υπάρχουν στο ευρύτερο πλαίσιο ACO άλλα και σε συγκεκριμένες υλοποιήσεις καθώς ένας αλγόριθμος ACO είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν βάση επιλογών που γίνονται στο σχεδιασμό του αλγορίθμου και των παραμέτρων που σταθμίζουν την επιρροή αυτών.

Οι εφαρμογές των αλγορίθμων σε στιγμιότυπα προβλήματα βελτιστοποίησης παρουσιάζονται στα κεφάλαια 6 και 7 ενώ στο κεφάλαιο 8 παρουσιάζεται ένας νέος αλγόριθμος που αντλεί έμπνευση από το πλαίσιο αλγορίθμων Ant Colony Optimization και προτείνει μια καινοτόμο διαδικασία ανανέωσης της φερομόνης με σκοπό τη ταχύτερη και καλύτερη απόδοση.

1.2 Στόχοι της Έρευνας

- Εκτενής έρευνα της διεθνούς αρθρογραφίας και βιβλιογραφίας με σκοπό την επιλογή των βέλτιστων αλγορίθμων και τη κατανόηση της συμπεριφοράς των αλγορίθμων σε σχέση με τα υπό έρευνα προβλήματα.

- Εφαρμογή διαφορετικών μεταευριστικών μεθόδων σε καινοτόμα προβλήματα.

- Έρευνα για την επιλογή των κατάλληλων τμημάτων των αλγορίθμων και παραμέτρων με βάση την απόδοση τους στο υπό έρευνα πρόβλημα.

- Πρόταση μιας νέας διαφοροποιημένης μορφής αλγορίθμου αποικίας μυρμηγκιών

- Έλεγχος της προτεινόμενης νέας διαφοροποιημένης μορφής αλγορίθμου αποικίας μυρμηγκιών με εφαρμογή του σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης.

1.3 Οργάνωση της Διατριβής

Κεφάλαιο 2

Παρουσιάζονται εκτενώς η θεωρία και στιγμιότυπα προβλημάτων συνδυαστικής βελτιστοποίησης. Πραγματικά προβλήματα με πλήθος εφαρμογών μπορούν να αναχθούν σε αυτά τα προβλήματα της κλάσης NP Complete και οι τροποποιήσεις τους με σκοπό να καλύψουν μεγαλύτερο εύρος εφαρμογών δείχνουν το ενδιαφέρον που έχουν προκαλέσει στη ερευνητική κοινότητα καθώς και την αντίστοιχη προσπάθεια που έχει γίνει για την επίλυσή τους. Το κεφάλαιο περιέχει αναλυτική μελέτη προβλημάτων συνδυαστικής βελτιστοποίησης που ερευνήθηκαν στην παρούσα διατριβή όπως: το Πρόβλημα Περιοδεύοντος Πωλητή, το Πρόβλημα Δρομολόγησης Οχημάτων, το Πρόβλημα Τετραγωνικής Ανάθεσης υπό τη μορφή του Προβλήματος Γραμμικής Τοποθέτησης Μηχανών. Το πρόβλημα του περιοδεύοντος πωλητή έχει απασχολήσει πλήθος ερευνητών λόγο της απλής έκφρασης του αλλά και της μεγάλης εφαρμογής του, όπως και το σχετικά συναφές από εννοιολογική άποψη πρόβλημα δρομολόγησης οχημάτων . Επίσης το πρόβλημα τετραγωνικής ανάθεσης και η σχέση του με τις διαδικασίες παραγωγής έχουν προσελκύσει πολλές ερευνητικές προσπάθειες.

Κεφάλαιο 3

Στο κεφάλαιο αυτό γίνεται μια πρώτη γνωριμία με τα αρχικά βήματα στην κατανόηση της συλλογικής συμπεριφοράς των συστημάτων ευφυΐας σμήνους. Αναλύονται οι πρώτες ιδέες για την αυτο-οργάνωση, τη στιγμεργεία, τη φερομόνη και τα πρώιμα μαθηματικά μοντέλα. Τα τέσσερα βασικά στοιχεία της αυτο-οργάνωσης, όπως η θετική ανάδραση, αρνητική ανάδραση, ενίσχυση τυχαίων διακυμάνσεων, αλλά και η ύπαρξη αλληλεπιδράσεων μεταξύ των ατόμων της αποικίας, δημιουργούν ένα σύστημα όπου το μεμονωμένο έντομο μπορεί να διεκπεραιώσει μόνο απλές λειτουργίες και δε λαμβάνεται υπόψη η περίπλοκη δομή του μεμονωμένου εντόμου. Η στιγμεργία είναι ένας αποκεντρωμένος μηχανισμός ροής πληροφοριών στο οποίο τα άτομα, μέρη του συστήματος, επικοινωνούν μέσω των αλλαγών που προκαλούν στο τοπικό τους περιβάλλον και στα κοινωνικά έντομα όπως τα μυρμήγκια υλοποιείται με την εναπόθεση στο έδαφος της χημικής ουσίας που ονομάζεται φερομόνη.

Τελικά παρουσιάζεται η μετάβαση από το βιολογικό στο τεχνητό μοντέλο συμπεριφοράς αποικίας τεχνητών μυρμηγκιών και γίνεται σχολιασμός του. Με την ωρίμανση της μαθηματικής απεικόνισης αυτών των ιδεών γεννήθηκαν οι αρχικοί αλγόριθμοι μυρμηγκιών και παρουσιάστηκαν οι πρώτοι απτοί αλγόριθμοι που αντλούσαν την έμπνευση για τον τρόπο λειτουργίας τους από τον τρόπο λειτουργίας των κοινωνικών εντόμων που παρατηρήθηκε στις αποικίες μυρμηγκιών.

Κεφάλαιο 4

Το κεφάλαιο δίνει μια εισαγωγή στην ευριστική και μεταευριστική έρευνα καθώς και στο γενικότερο πλαίσιο της μεταευριστικής μεθόδου Ant Colony Optimization. Τα κύρια σημεία ενός αλγορίθμου μυρμηγκιών είναι : ο πίνακας φερομόνης, ο πιθανοκρατικός κανόνας κατασκευής λύσεων, ο κανόνας ανανέωσης της φερομόνης, η ευριστική πληροφορία και η προαιρετική ενσωμάτωση daemon action συνήθως με τη μορφή μιας μεθόδου τοπικής έρευνας. Παρουσιάζονται εκτενώς τα μέλη της οικογένειας μεταευριστικών αλγορίθμων Ant Colony Optimization καθώς και οι εφαρμογές τους από τη διεθνή αρθρογραφία έτσι ώστε να γίνεται εκτίθεται ο σχεδιασμός του αλγορίθμου σε συνδυασμό με το υπό έρευνα πρόβλημα.

Συνοπτικά, παρατηρούμε ότι οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών ACO διαθέτει αρκετά χαρακτηριστικά που όταν συνδυάζονται δημιουργούν καινοτόμες προσεγγίσεις : χρησιμοποιεί ένα *πληθυσμό* (αποικία) τεχνητών μυρμηγκιών που *κατασκευάζουν* λύσεις εκμεταλλευόμενα την *έμμεση μορφή μνήμης* που είναι οι *τεχνητές φερομόνες.*

Κεφάλαιο 5

Το κεφάλαιο αυτό παρουσιάζει εναλλακτικές μεθόδους προσέγγισης των δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης όπως η μέθοδος Προσομοιωμένης Ανόπτησης, οι Γενετικοί Αλγόριθμοι, η Tabu Search και η API. Οι εναλλακτικές μεθοδολογίες που παρουσιάζονται στη παρούσα εργασία είναι αλγόριθμοι οι οποίοι μπορούν να εφαρμοστούν είτε μόνες τους είτε ως μέθοδοι τοπικής έρευνας και στη διεθνή βιβλιογραφία υπάρχουν δημοσιεύσεις όπου οι μέθοδοι υλοποιούνται είτε με τον ένα είτε με τον άλλο τρόπο, αλλά όχι πάντα με επιτυχημένα αποτελέσματα. Ο συνδυασμός τους με το πλαίσιο ACO συνήθως έχει τη μορφή τη τοπικής έρευνας και αυξάνει τη σημασία του σωστού σχεδιασμού και των σωστών παραμέτρων στο αλγόριθμο μυρμηγκιών.

Κεφάλαιο 6

Παρουσιάζεται η έρευνα ενός προβλήματος συνδυαστικής βελτιστοποίησης για τη βελτιστοποίηση θαλάσσιων μεταφορών με τον αλγόριθμο προσομοιωμένης ανόπτησης επιλέγοντας ένα σύνολο δεκατριών νησιών του Αιγαίου για ένα πρόβλημα περιοδεύοντος πωλητή. . Ο κύριος ρυθμιστικός παράγοντας της συμπεριφοράς του αλγορίθμου είναι το χρονοπρόγραμμα ψύξης. Ένα σχετικά γρήγορο χρονοπρόγραμμα ψύξης, όπως και στο φυσικό μοντέλο ανόπτησης μετάλλων, δεν θα δώσει στα μόρια/τμήματα λύσης το χρόνο που χρειάζονται ώστε να διαταχθούν με την ελάχιστη ενέργεια/κόστος. Όμως, ένα πολύ αργό πρόγραμμα ψύξης θα αυξήσει δυσανάλογα τον υπολογιστικό χρόνο που χρειάζεται για την παραγωγή λύσεων. Σημαντικό επίσης είναι το κριτήριο αποδοχής νέων λύσεων καθώς σε αυτό το σημείο κρίνεται η επιλογή μεταξύ εξερεύνησης νέων λύσεων και εκμετάλλευσης παλαιότερων. Αποτελέσματα υψηλής ποιότητας βασίζονται στη σωστή επιλογή αυτών των κριτηρίων στο υπό έρευνα πρόβλημα και ουσιαστικά από αυτά εξαρτάται η ευρωστία του αλγορίθμου.

Κεφάλαιο 7

Το κεφάλαιο 7 ασχολείται με εφαρμογές του αλγορίθμου Ant Colony System για την επίλυση ενός προβλήματος δρομολόγησης οχημάτων, πρόβλημα βέλτιστης διαχείρισης θαλάσσιων μεταφορών με περιορισμό χωρητικότητας καθώς και πρόβλημα διάταξης μηχανών σε κατασκευαστική κυψέλη. Η εφαρμογή του Ant Colony System σε αυτά τα προβλήματα απέδωσε ικανοποιητικά αποτελέσματα αποδεικνύοντας την προσαρμοστικότητα και αποτελεσματικότητα του πλαισίου αλγορίθμων μυρμηγκιών καθώς τα υπολογιστικά αποτελέσματα δείχνουν ότι το πλαίσιο του αλγορίθμου Ant Colony System είναι μια αποτελεσματική μεταευριστική μέθοδος. Όμως, γενικά για μια εφαρμογή στους αλγορίθμους ACO, η δυσκολία έγκειται στο προσδιορισμό του σωστού σχεδιασμού και στη ρύθμιση του πλήθους των παραμέτρων με σκοπό τη βέλτιστη απόδοση στο υπό έρευνα πρόβλημα. Η επιλογή της σωστής περιοχής τιμών των παραμέτρων είναι ζήτημα πειραματισμού με την εφαρμογή.

Κεφάλαιο 8

Προτείνεται μια νέα προσέγγιση του αλγορίθμου Ant Colony System και δοκιμάζεται με εφαρμογή σε πρόβλημα περιοδεύοντος πωλητή για 100,500 και 1000 πόλεις. Ο νέος αλγόριθμος αντλεί έμπνευση από το πλαίσιο αλγορίθμων Ant Colony Optimization και προτείνει μια νέα διαδικασία ανανέωσης της φερομόνης με σκοπό τη ταχύτερη και καλύτερη απόδοση. Η λειτουργία του αλγορίθμου δοκιμάζεται στα τρία στιγμιότυπα του προβλήματος με ένα σύνολο αρχικών παραμέτρων και στη συνέχεια ελέγχεται η απόδοση του με αλλαγή των παραμέτρων που επηρεάζουν τη λειτουργία των τμημάτων του. Ως αρχικές τιμές για τις παραμέτρους της διαδικασίας χρησιμοποιούνται τιμές κοντά στις πειραματικά προτεινόμενες από τη διεθνή βιβλιογραφία και στη συνέχεια ελέγχονται οι διαδικασίες κατασκευής λύσεων, ανανέωσης της φερομόνης με νέες τιμές για τις παραμέτρους. Τα αποτελέσματα είναι ικανοποιητικά και ο νέος αλγόριθμος δείχνει τη σημασία του σωστού σχεδιασμού, της σωστής ρύθμισης των παραμέτρων για την επίτευξη του βέλτιστου αποτελέσματος καθώς και την προσαρμοστικότητα του πλαισίου αλγορίθμων μυρμηγκιών.

Κεφάλαιο 9

Παρουσιάζονται τα συμπεράσματα της έρευνας και σχολιάζονται τα ευρήματα της. Τέλος προτείνονται μελλοντικές περιοχές έρευνας.

Κεφάλαιο 2

Προβλήματα Συνδυαστικής Βελτιστοποίησης

2.1 Εισαγωγή

Τα πεδία εφαρμογής των Προβλημάτων Συνδυαστικής Βελτιστοποίησης (Combinatorial Optimization Problems) είναι ιδιαίτερα σημαντικά καθώς αφορούν τις μεταφορές τις τηλεπικοινωνίες και τις βιομηχανικές κατασκευαστικές διαδικασίες. Η υπολογιστική ισχύς υπάρχει πλέον παντού και η συνεχής αύξησή της μας επιτρέπει την προσομοίωση πραγματικών προβλημάτων με σκοπό την αναζήτηση καλύτερων λύσεων μέσω αλγοριθμικής επίλυσής τους. Το συγκριτικό πλεονέκτημα που μπορεί να αποκτηθεί με τέτοιες λύσεις είναι χαμηλότερο κόστος σε σχέση με τους ανταγωνιστές το οποίο προέρχεται από την επιστημονική βάση προσέγγισης του προβλήματος.

Με ένα μη αυστηρό ορισμό, ένα πρόβλημα συνδυαστικής βελτιστοποίησης είναι ένα πρόβλημα η λύση του οποίου συντίθεται από μια σειρά από *αντικείμενα*. Για κάθε αντικείμενο υπάρχει μια επιλογή από *τιμές*. Αυτός ο συνδυασμός *αντικειμένου-τιμής* αποτελεί μια λύση η οποία μπορεί να βαθμολογηθεί με βάση κάποια χαρακτηριστικά που ορίζονται από τον ερευνητή. Όταν η βαθμολογία αυτή, η οποία καλείται αντικειμενική συνάρτηση (objective function), μεγιστοποιείται ή ελαχιστοποιείται ανάλογα με τη φύση του προβλήματος, θεωρούμε ότι το πρόβλημα λύθηκε.

Πιο αυστηρή μαθηματική έκφραση ενός προβλήματος συνδυαστικής βελτιστοποίησης δίδεται από τους Papadimitriou και Steiglitz[1]. Το γενικό μη-γραμμικής μορφής πρόβλημα προγραμματισμού (general nonlinear programming problem) ορίζεται ως:

Προσδιορίστε το x με σκοπό τη ελαχιστοποίηση της $f(x)$

$$\min f(x)$$

με $\quad g_i(x) \geq 0 \quad$ και $\quad i = 1,\dots,m$

$$h_j(x) = 0 \quad \text{και} \quad j = 1,\dots,p$$

όπου f, g_i, h_j συναρτήσεις της μεταβλητής $x \in R^n$.

Όταν η f είναι κυρτή (convex), η g_i κοίλη (concave) και η h_j γραμμική (linear), το πρόβλημα ονομάζεται πρόβλημα κυρτού προγραμματισμού (convex programming problem). Σε αυτή τη κατηγορία το τοπικό βέλτιστο είναι και το ολικό βέλτιστο.

Όταν η f, η g_i και η h_j είναι γραμμικές το πρόβλημα ονομάζεται πρόβλημα γραμμικού προγραμματισμού (linear programming problem). Σε αυτή τη κατηγορία προβλημάτων η βελτιστοποίηση γίνεται με την επιλογή μιας λύσης από ένα πεπερασμένο σύνολο λύσεων και αυτό το πρόβλημα ονομάζεται συνδυαστικό.

[1] Papadimitriou C. , Steiglitz K. (1982) Combinatorial Optimization: Algorithms and Complexity, Prentice Hall ,New Jersey.

Πρόβλημα Βελτιστοποίησης και Πρόβλημα Απόφασης

Όποιο πρόβλημα έχει ως σκοπό την εύρεση μιας βέλτιστης (είτε ελάχιστης, είτε μέγιστης) τιμής μιας δοσμένης συνάρτησης κόστους καλείται πρόβλημα βελτιστοποίησης και αντίστοιχα ο αλγόριθμος επίλυσης του καλείται αλγόριθμος βελτιστοποίησης. Επίσης, όποιο πρόβλημα έχει ως λύση είτε το *μηδέν*, είτε το *ένα* ή εναλλακτικά ως λύση ένα *ναι* ή ένα *όχι*, ονομάζεται πρόβλημα απόφασης και αντίστοιχα ο αλγόριθμος επίλυσης του καλείται αλγόριθμος απόφασης. Τα προβλήματα βελτιστοποίησης μπορούν να εκφραστούν ως προβλήματα απόφασης αν θέσουμε κάποιον όρο και συνήθως εάν υπάρχει ένας γρήγορος αλγόριθμος για ένα πρόβλημα απόφασης, είναι σχετικά εύκολο να λυθεί και το αντίστοιχο πρόβλημα βελτιστοποίησης.

Η συνδυαστική έκρηξη είναι η εκθετική αύξηση του μεγέθους του χώρου αναζήτησης του προβλήματος και είναι και ο κύριος λόγος που αυτά τα προβλήματα με τη συνήθως μικρή και κομψή μαθηματική έκφραση είναι τόσο δύσκολα να λυθούν και συχνά κατατάσσονται στη κλάση NP (Non deterministic Polynomial). Για παράδειγμα στο ασύμμετρο πρόβλημα του περιοδεύοντος πωλητή ο χώρος αναζήτησης είναι μεγέθους $(n-1)!$ με n των αριθμό των πόλεων. Αυτό σημαίνει ότι τέτοιου είδους προβλήματα μπορούν να εξακριβωθούν (verified) αλλά όχι να λυθούν σε πολυωνυμικό χρόνο.

Μέσα στη κλάση NP υπάρχουν δύο σχετιζόμενα σύνολα προβλημάτων, τα NP-Complete προβλήματα και τα NP-Hard προβλήματα. NP-Hard προβλήματα είναι όλα τα προβλήματα τα οποία μπορούν να αναχθούν (reduced) σε προβλήματα απόφασης στην NP με μια πολυωνυμική μέθοδο όπως έδειξε ο Karp[2]. Εάν ένα πρόβλημα ανήκει στη κλάση NP και είναι NP-Hard τότε καλείται NP-Complete.

P Πρόβλημα

P είναι το σύνολο όλων των προβλημάτων απόφασης (decision problems), που μπορούν να λυθούν από ντετερμινιστικούς αλγορίθμους σε πολυωνυμικό χρόνο. Δηλαδή, λέμε ότι ένα πρόβλημα ανήκει στη κλάση P εάν υπάρχει ένας αλγόριθμος A και ένας αριθμός c,τέτοιος ώστε για κάθε έκφραση (Instance) του προβλήματος I ο αλγόριθμος A θα παράγει τη λύση σε χρόνο $O(B^c)$, όπου B είναι ο αριθμός των bit στο αλφαριθμητικό (input string) που αναπαριστά το I.

Για τη μαθηματική έκφραση του μοντέλου χρησιμοποιείται η ντετερμινιστική μηχανή Turing (Deterministic Turing Machine) η οποία προτάθηκε από τον Allan Turing το 1936 πριν την κατασκευή υπολογιστών. Αφού κάθε πρόβλημα απόφασης μπορεί να έχει μόνο δύο απαντήσεις (*ναι* ή *όχι*) , μπορούμε να θέσουμε ένα πρόβλημα απόφασης ρωτώντας αν μια δοσμένη λέξη (input string) ανήκει ή δεν ανήκει σε μία συγκεκριμένη γλώσσα. Η γλώσσα είναι το σύνολο των λέξεων (input string) για τις οποίες η απάντηση είναι *ναι*.

Τα στοιχεία της κλάσης P είναι γλώσσες (languages). Ορίζουμε τη κλάση γλωσσών P ως:

$$P = \{L \mid L = L(M) \text{ για μια μηχανή Turing σε πολυωνιμικό χρόνο}\}$$

Όπου :

[2] Karp, R. M. (1972). Reducibility among combinatorial problems. In Complexity of Computer Computations, σελ. 85.103. Plenum Press.

$L \subseteq \Sigma^*$ με Σ^* το πεπερασμένο σύνολο συμβολοσειράς του Σ, ενός πεπερασμένου αλφαβήτου με τουλάχιστον δύο σύμβολα.

M μια μηχανή Turing

$L(M)$ η γλώσσα της μηχανής M με το αλφάβητο Σ και ορίζεται ως

$L(M) = \{w \in \Sigma^* \mid \text{M δέχεται } w\}$ έτσι ώστε η M δέχεται το w ,μια συμβολοσειρά του Σ^*, αν τερματίζει τους υπολογισμούς της σε μια αποδεκτή κατάσταση(state).

NP Πρόβλημα

NP είναι το σύνολο όλων των προβλημάτων απόφασης (decision problems), που μπορούν να λυθούν από μη-ντερμινιστικούς αλγορίθμους σε πολυωνυμικό χρόνο. Κάθε πρόβλημα απόφασης στη P είναι και στην NP.

Πιο συγκεκριμένα, ορίζοντας ένα πρόβλημα απόφασης Q , το πρόβλημα αυτό ανήκει στη κλάση NP, αν υπάρχει ένας αλγόριθμος A που έχει τις παρακάτω ιδιότητες :

- Υπάρχει ένα *πιστοποιητικό (certificate)* $C(I)$ που σχετίζεται με κάθε λέξη της γλώσσας Q (δηλαδή με κάθε έκφρασή (instance) I για τη οποία η απάντηση είναι *'ναι'*) , ώστε όταν το ζεύγος $(I, C(I))$ εισάγονται στον αλγόριθμο A , αυτός αναγνωρίζει ότι η έκφραση I ανήκει στη γλώσσα Q.
- Εάν I μια λέξη που δεν ανήκει στη γλώσσα Q, τότε δεν υπάρχει αντίστοιχο *πιστοποιητικό (certificate)* $C(I)$ που θα κάνει τον A να αναγνωρίζει ότι η έκφραση I ανήκει στη γλώσσα Q.

* Ο αλγόριθμος A λειτουργεί σε πολυωνυμικό χρόνο.

Για να το θέσουμε συντομότερα, η κλάση NP είναι η κλάση προβλημάτων απόφασης για την οποία είναι εύκολο να *ελέγξουμε* (check) την ορθότητα της υποτιθέμενης λύσης, με τη βοήθεια της πληροφορίας που προέρχεται από το πιστοποιητικό. Δηλαδή δε ζητάμε τρόπο *εύρεσης* της λύσης *αλλά* μόνο τρόπο εξακρίβωσης (verify) ότι η υποτιθέμενη λύση είναι πράγματι αληθείς.

Για παράδειγμα, στο πρόβλημα του περιοδεύοντος πωλητή ένα πιστοποιητικό είναι μια διαδρομή (tour) της οποίας το συνολικό μήκος είναι $\leq K$. Ο αλγόριθμος A θα εξακριβώσει (verify) ότι πράγματι η διαδρομή επισκέπτεται όλες τις πόλεις και πράγματι έχει συνολικό μήκος $\leq K$.

NP-hard Πρόβλημα

NP-hard (Non-deterministic Polynomial) είναι το σύνολο όλων των προβλημάτων απόφασης, για τα οποία εάν ένας αλγόριθμος μπορεί να επιλύσει ένα πρόβλημα της κλάσης τότε μπορεί να επιλύσει και οποιοδήποτε πρόβλημα NP σε πολυωνυμικό χρόνο. Η κλάση *NP* είναι μια κλάση προβλημάτων απόφασης που είναι δυνατόν να λυθούν σε πολυωνυμικό χρόνο από μια μη-ντερμινιστική υπολογιστική μηχανή Turing. Επίσης η κλάση NP μπορεί να ελεγχθεί (verified) από μια ντετερμινιστική μηχανή Turing.

Ένα πρόβλημα C, είναι NP-Hard εάν $\forall C' \in NP$, το C' μπορεί να αναχθεί πολυωνυμικά (polynomial time reduction) σε ένα πρόβλημα τύπου C.

NP-complete Πρόβλημα

NP-Complete είναι το σύνολο των προβλημάτων C για τα οποία πρέπει : $C \in NP$ και είναι NP-Hard.

Το 1971 ο Cook[3] απέδειξε ότι ένα συγκεκριμένο πρόβλημα, το πρόβλημα ικανοποίησης (satisfiability problem) έχει την ιδιότητα, ότι οποιοδήποτε πρόβλημα στην NP μπορεί πολυωνυμικά να αναχθεί (reduced) σε αυτό. Συνεπώς αν αυτό μπορεί να λυθεί σε πολυωνυμικό χρόνο τότε όλα τα προβλήματα της NP μπορούν, άρα P=NP. Αλλά και αν ένα αποδειχθεί ότι δεν μπορεί να λυθεί σε πολυωνυμικό χρόνο τότε όλα δεν μπορούν να λυθούν. Αυτό το ερώτημα δεν έχει απαντηθεί και παραμένει το μεγαλύτερο ανοιχτό πρόβλημα της επιστήμης των υπολογιστών. Για να συγκρίνουμε την απόδοση των αλγορίθμων στη κλάση προβλημάτων P και αυτών στη NP , έχει δημιουργηθεί η έννοια του μη ντετερμινιστικού υπολογιστή (non-deterministic computer). Αυτό το μοντέλο υπολογιστικής μηχανής δεν υπάρχει ως πραγματικό αντικείμενο. Ένας κανονικός υπολογιστής είναι ντετερμινιστική υπολογιστική μηχανή, δηλαδή η κατάσταση (state) σε μια συγκεκριμένη χρονική στιγμή t , ορίζεται με προβλέψιμο τρόπο από την κατάσταση και την εισροή (input) της προηγούμενη χρονικής στιγμής t-1.

Λόγω της συνδυαστικής έκρηξης που συμβαίνει στο χώρο αναζήτησης, αυτά τα προβλήματα είναι πολύ δύσκολο να επιλυθούν με παραδοσιακούς αλγορίθμους που εξασφαλίζουν την απόλυτη βεβαιότητα της βέλτιστης λύσης καθώς θα χρειαζόταν τεράστιος υπολογιστικός χρόνος για να φθάσουμε σε αυτή. Για αυτό το λόγο η έρευνα στοχεύει να προσεγγίσει αυτά τα προβλήματα με προσεγγιστικούς αλγορίθμους οι οποίοι είναι πολύ πιο γρήγοροι και βρίσκονται πάρα πολύ κοντά στην απόλυτα βέλτιστη λύση.

[3] Cook S. A. (1971) "The complexity of theorem proving procedures". Proceedings, Third Annual ACM Symposium on the Theory of Computing, ACM, New York, p. 151-158

2.2 Το Πρόβλημα του Περιοδεύοντος Πωλητή

Η κλασική περιγραφή του Προβλήματος του Περιοδεύοντος Πωλητή (Travelling Salesman Problem/TSP) είναι η παρακάτω: Ένας πωλητής πρέπει να επισκεφθεί n πόλεις, που απεικονίζονται με τους αριθμούς $1, 2, ..., n$, ξεκινώντας από μια αρχική πόλη που θεωρούμε ως βάση, την πόλη 1, *αλλά* πρέπει να επισκεφθεί όλες τις υπόλοιπες $n-1$ πόλεις ακριβώς μία φορά και να επιστρέψει στη πόλη 1. Σκοπός είναι η δημιουργία μιας διαδρομής που ελαχιστοποιεί την απόσταση που χρειάζεται να ταξιδέψει ο πωλητής.

Με όρους της θεωρίας γραφημάτων, αυτό το πρόβλημα είναι ανάλογο με την εύρεση ενός ελάχιστου Χαμιλτονιανού κυκλώματος (Hamiltonian circuit) σε πλήρες γράφημα $K_n \equiv G(N, A)$, όπου $N = \{1, 2, ..., n\}$ είναι το σύνολο των κόμβων/κορυφών (nodes,vertices) και $A = \{(i, j) : i, j \in N, i \neq j\}$ το σύνολο των τόξων που συνδέουν πλήρως τους κόμβους του συνόλου N. Ο πίνακας για το κόστος μετάβασης ή αποστάσεων θεωρείται θετικός, $C = [c_{ij}]$ όπου c_{ij} το κόστος μετάβασης ή απόσταση μεταξύ του κόμβου i και j. Η κατεύθυνση διάσχισης των κόμβων είναι σημαντική για το πρόβλημα. Όταν $c_{ij} \neq c_{ji}$ το πρόβλημα καλείται ασύμμετρο (asymmetric), ενώ όταν $c_{ij} = c_{ji}$ καλείται συμμετρικό (symmetric).

Μαθηματική Έκφραση του Προβλήματος του Περιοδεύοντος Πωλητή

Το 1954 οι Dantzig, Fulkerson και Johnson[4] εκφράζουν την γενική ασύμμετρη περίπτωση του Προβλήματος του Περιοδεύοντος Πωλητή. Θέτουν τη δυαδική μεταβλητή x_{ij} που ορίζεται ως:

[4] Dantzig, G.B., Fulkerson, D.R., Johnson, S., 1954. Solution of a large-scale Traveling Salesman Problem. Operations Research, 2, 393–410

$$x_{ij} = \begin{cases} 1, \text{ εάν το τόξο ij ανήκει στη λύση} \\ 0, \text{ αλλιώς} \end{cases} \tag{1}$$

Επίσης, πρέπει κάθε κόμβος να έχει ακριβώς ένα τόξο που κατευθύνεται προς αυτόν και ακριβώς ένα τόξο που κατευθύνεται από αυτόν σε αντίθετη κατεύθυνση. Το πρόβλημα των υπο-γύρων (subtours), με άλλα λόγια της εμφάνισης ασύνδετων βρόχων (disjointed loops) του γραφήματος, αντιμετωπίζεται με την παρακάτω έκφραση του προβλήματος.

$$\min \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij} \tag{2}$$

με
$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, ..., n \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, ..., n \tag{4}$$

και

$$\sum_{i,j \in S}^{n} x_{ij} \le |S| - 1 \text{ με } S \subset \{1, ..., n\} \tag{5}$$

$$x_{ij} = \{0, 1\}, \quad \forall \text{i,j και } i \neq j \tag{6}$$

όπου S ένα υποσύνολο των κόμβων/κορυφών/πόλεων.

Η εξίσωση της αντικειμενικής συνάρτησης (2) εκφράζει το κόστος της βέλτιστης διαδρομής/γύρου (tour). Οι εξισώσεις (3) και (4) ορίζουν ότι κάθε κόμβος/κορυφή/πόλη *ij* θα εισαχθεί στη λύση ακριβώς μια φορά και είναι περιορισμοί βαθμού (degree constraints). Για το πρόβλημα των υπογύρων προτείνεται η εξίσωση (5) που είναι περιορισμός με σκοπό την εξάλειψη των υπογύρων (subtour elimination constraint). Η παραπάνω διατύπωση, έχει $n(n-1)$ δυαδικές μεταβλητές, $2n$ περιορισμούς βαθμού (degree constraints) και $2^n - 2n - 2$ περιορισμούς εξάλειψης υπογύρου (subtour elimination constraint). Συνεπώς ακόμα και μικρές τιμές του n δημιουργούν ένα δυσεπίλυτο πρόβλημα.

Οι Miller, Tucker και Zemlin[5] προτείνουν μια διατύπωση που μειώνει τους περιορισμούς για την εξάλειψη των υπογύρων με τον ορισμό των μεταβλητών u_i, $i = 2,...,n$ όπου οι περιορισμοί γίνονται:

$$u_i - u_j + (n-1)x_{ij} \leq n-2 \qquad \text{με} \qquad i, j = 2,...,n \quad \text{και} \quad i \neq j \qquad (8)$$

$$1 \leq u_i \leq n-1 \qquad \text{με} \qquad i = 2,...,n \qquad (9)$$

Η εξίσωση (9) ορίζει ότι οι μεταβλητές u_i σε οποιοδήποτε γύρο είναι μοναδικές. Η εξίσωση (8) ορίζει ότι η προτεινόμενη διαδρομή δεν περιέχει κανένα υπογύρο σε συνολο κορυφών με $S \subseteq V \setminus \{1\}$, άρα δεν περιέχει κανένα υπογύρο που έχει λιγότερες από n κορυφές.

Στη συμμετρική εκδοχή του προβλήματος του περιοδεύοντος πωλητή (S-TSP) ισχύει ότι $c_{ij} = c_{ji}$ άρα η κατεύθυνση διάσχισης των κόμβων του γραφήματος δεν επηρεάζει το κόστος/μηκος της διαδρομής λύσης του προβλήματος. Συνεπώς θεωρούμε ένα μη κατευθυνόμενο γράφημα (undirected graph) όπου μόνο ένα τόξο συνδέει δύο κόμβους. Τότε ορίζουμε τη μεταβλητή απόφασης $x_j = \{0,1\}$, όπου j αφορά όλες τις ακμές (edge) E του μη κατευθυνόμενου γραφήματος και c_j είναι το κόστος διάσχισης της ακμής j. Η εύρεση μιας

[5] Miller C.E., Tucker A.W., Zemlin R.A.,(1960) Integer programming formulation of the traveling salesman problem. Journal of Association Computing Machinery, 7:4 p 326-329.

διαδρομής λύση πρέπει να επιλεγεί από ένα υποσύνολο των ακμών έτσι ώστε κάθε κόμβος να περιέχεται σε ακριβώς δύο από τις ακμές που έχουμε επιλέξει. Οι παρακάτω συναρτήσεις εκφράζουν το συμμετρικό πρόβλημα.

$$\min \frac{1}{2} \sum_{j=1}^{n} \sum_{k \in J(j)} c_k x_k$$

με $$\sum_{k \in J(j)} x_k = 2 \qquad \forall j = 1, 2, \ldots, n$$

$$\sum_{i \in E(S)} x_i \leq |S| - 1 \qquad \forall S \subset \{1, 2, \ldots, n\}$$

$$x_j = \{0, 1\} \qquad \forall j \in E(S)$$

όπου $J(j)$ το σύνολο των μη κατευθυνόμενων ακμών (undirected edges) που συνδέονται στο κόμβο j και $E(S)$ το υποσύνολο των μη κατευθυνόμενων ακμών που συνδέουν τις πόλεις που ανήκουν στο υποσύνολο των πόλεων S, με εφικτό τρόπο ώστε να μην δημιουργούνται υπογύροι. Σε αυτή την έκφραση του προβλήματος υπάρχουν $n(n-1)/2$ δυαδικές μεταβλητές, δηλαδή οι μισές της ασύμμετρης περίπτωσης. Αν και η συμμετρική περίπτωση του προβλήματος μπορεί να θεωρηθεί ως μια ειδική περίπτωση της ασύμμετρης, η εμπειρία έχει δείξει ότι οι αλγόριθμοι επίλυσης της ασύμμετρης περίπτωσης, εν γένει , δεν αποδίδουν το ίδιο καλά στη συμμετρική.

Εφαρμογές του Προβλήματος του Περιοδεύοντος Πωλητή

Η μαθηματική μοντελοποίηση του προβλήματος μπορεί να εφαρμοστεί σε πλήθος πρακτικών προβλημάτων. Με την τυπική μορφή της εύρεσης της συντομότερης και μικρότερου κόστους διαδρομής όλων των πόλεων, μπορούμε να το εφαρμόσουμε στις μεταφορές ή στη χάραξη της φυσικής διαδρομής δικτύων επικοινωνιών. Πάνω στο τυπικό πρόβλημα όμως, με κάποιους πρόσθετους περιορισμούς, είναι δυνατή η μοντελοποίηση των προβλημάτων δρομολόγησης (routing problems) και προβλημάτων μεταθέσεων. Προβλήματα με δομή TSP, προκύπτουν στο σχεδιασμό τυπωμένων κυκλωμάτων (printed circuits) όπου πρέπει να ανοιχτεί ένας συγκεκριμένος αριθμός από τρύπες σε τυπωμένες πλακέτες. Με ένα ρομποτικό τρυπάνι, που ξεκινά από συγκεκριμένο σημείο, έχει ως σκοπό να ακολουθήσει μια διαδρομή όπου οι κόμβοι της διαδρομής είναι οι τρύπες που πρέπει να ανοιχθούν. Μια συντομότερη διαδρομή θα σημαίνει επεξεργασία περισσότερων πλακετών στη μονάδα του χρόνου.

Στη ανάλυση της δομής των κρυστάλλων χρειάζεται να ληφθούν μια σειρά από ακτινογραφίες. Μετά την τοποθέτηση των δειγμάτων στο μηχάνημα γίνεται μια σειρά από μετρήσεις στο κρύσταλλο. Η σειρά με την οποία πραγματοποιούνται οι διάφορες μετρήσεις σε ένα δεδομένο κρύσταλλο μπορεί να θεωρηθεί ως μια λύση ενός TSP προβλήματος. Επίσης τέτοιου τύπου προβλήματα εμφανίζονται στο χειρισμό υλικών σε αποθήκες και στον ορισμό ακολουθίας μηχανών για την παραγωγή προϊόντων όπου μόνο μια μηχανή χρησιμοποιείται.

Τροποποιήσεις του Προβλήματος του Περιοδεύοντος Πωλητή

Η αρχική περιγραφή του προβλήματος που δίδεται παραπάνω, αποτέλεσε το γεννήτορα διαφόρων τροποποιημένων εκδοχών.

Πολλαπλό Πρόβλημα Περιοδεύοντος Πωλητή

Το Πολλαπλό Πρόβλημα Περιοδεύοντος Πωλητή (Multiple TSP) είναι μια εκδοχή του κλασικού προβλήματος όπου είναι δυνατό να χρησιμοποιηθούν περισσότεροι του ενός πλανόδιοι πωλητές, όπου όλοι ξεκινούν και επιστρέφουν στην ίδια πόλη αφετηρίας. Καλύπτει μια ευρεία κατηγορία προβλημάτων και με μερικές τροποποιήσεις στους περιορισμούς είναι δυνατό να περιγράψει προβλήματα δρομολόγησης οχημάτων (Vehicle Routing Problems). Το Πολλαπλό Πρόβλημα Περιοδεύοντος Πωλητή δεν έχει προσελκύσει πλήθος ερευνητών, παρά το γεγονός ότι τα προβλήματα δρομολόγησης και το κλασικό πρόβλημα περιοδεύοντος πωλητή έχουν μελετηθεί πολύ και θεωρούνται προβλήματα αναφοράς.

Το Γενικευμένο Πρόβλημα Περιοδεύοντος Πωλητή

Το Γενικευμένο Πρόβλημα Περιοδεύοντος Πωλητή (Generalized TSP) παρουσιάστηκε από τον Labordere[6]. Είναι ειδική περίπτωση του κλασικού ΠΠΠ όπου ένας πλανόδιος πωλητής πρέπει να περάσει από έναν προκαθορισμένο αριθμό υποσυνόλων πελατών, επισκεπτόμενος τουλάχιστον έναν πελάτη από κάθε υποσύνολο, ελαχιστοποιώντας ταυτόχρονα το ποσό των δαπανών επίσκεψης (sum of traveling costs). Συνεπώς πρέπει να επιλέξουμε τη σειρά με την οποία θα επισκεφθεί ο πωλητής τα υποσύνολα πελατών, αλλά και τον πελάτη ή τους πελάτες που πρέπει να επισκεφτεί σε κάθε υποσύνολο. Το πρόβλημα μπορεί να οριστεί τόσο για κατευθυνόμενα όσο και για μη κατευθυνόμενα γραφήματα, συμμετρικά ή μη και η πιο συχνή εκδοχή αυτού του προβλήματος που συναντάμε στη βιβλιογραφία είναι αυτή όπου ο πωλητής πρέπει να επισκεφτεί ακριβώς έναν πελάτη σε κάθε υποσύνολο. Ένα μεγάλο πλήθος συνδυαστικών προβλημάτων βελτιστοποίησης μπορούν να μοντελοποιηθούν με βάση αυτό το πρόβλημα και ο Laporte[7] αναφέρει ενδεικτικά τα προβλήματα δρομολόγησης θέσης (location routing problems), μερικά προβλήματα σχεδιασμού συστημάτων ροής υλικού (material-flow system design problems), το

[6] Labordere H. (1969) The record balancing problem: A dynamic programming solution of a generalized traveling salesman problem, RIBO B 2, 43-49.

[7] Laporte, G.(1992) The vehicle routing problem: An overview of exact and approximate algorithms, Journal European Journal of Operational Research, Vol 59 Issue 3 : 345-358

πρόβλημα αποκομιδής ταχυδρομικών κουτιών (post-box collection problem) και το πρόβλημα δρομολόγησης τόξων (arc routing problem).

Το Πρόβλημα Περιοδεύοντος Πωλητή με Συλλογή Βραβείων

Το Πρόβλημα Περιοδεύοντος Πωλητή με Συλλογή Βραβείων (Prize Collecting TSP) παρουσιάστηκε αρχικά από τους Balas & Martin[8] και αργότερα το ερεύνησαν οι Fischetti & Toth[9]. Το πρόβλημα ορίζει ότι ένας πωλητής ταξιδεύει από τον έναν πελάτη στον άλλο με δεδομένο κόστος, πληρώνει ποινική ρήτρα (penalty) για κάθε πελάτη που αποτυγχάνει να επισκεφτεί, ενώ συλλέγει ένα βραβείο (prize) για κάθε πελάτη που επισκέπτεται. Σκοπός του προβλήματος είναι ο πωλητής να συλλέξει ένα χρηματικό ποσό από τα βραβεία μεγαλύτερο ή ίσο από ένα δεδομένο ποσό, ελαχιστοποιώντας ταυτόχρονα το ποσό των εξόδων για τη διαδρομή (traveling costs) αλλά και των ποινικών ρητρών (penalty costs) που πρέπει να πληρώσει.

Το Πιθανοθεωρητικό Πρόβλημα Περιοδεύοντος Πωλητή

Το Πιθανοθεωρητικό Πρόβλημα Περιοδεύοντος Πωλητή (Probabilistic TSP) προτείνεται από την Jaillet[10]. Βασίζεται και αυτό στο κλασικό ΠΠΠ αλλά σε αυτό το πρόβλημα ένας πωλητής μπορεί να επισκεφθεί μόνο ένα υποσύνολο κόμβων σε οποιαδήποτε δεδομένη στιγμή του προβλήματος. Μια δεδομένη κατανομή πιθανότητας ορίζει το ενεργό αυτό υποσύνολο κόμβων και ο σκοπός είναι η εύρεση ενός a priori γύρου με το ελάχιστο αναμενόμενο μήκος, για όλα τα υποσύνολα κόμβων, με βάση τη στρατηγική επίσκεψης όλων των ενεργών κόμβων, του

[8] Balas E., Martin C., The prize collecting travelling salesman problem. Networks Volume 19, Issue 6 , Pages 621 – 636 (1989) Willey periodicals.

[9] Fischetti M., Toth P. (1988) An additive approach for the optimal solution of the prize collecting traveling salesman problem. In Vehicle Routing Methods ad Studies. , North Holland p 319-343.

[10] Jaillet P., (1985) The Probabilistic Traveling Salesman Problems. Technical Report Center MIT Cambirdge Mass.

υποσυνόλου, με την ίδια σειρά όπως εμφανίζονται στον a priori γύρο. Το αναμενόμενο μήκος υπολογίζεται με βάση την πιθανότητα εμφάνισης και το συνολικό κόστος του εκάστοτε γύρου. Τα καινοτόμα στοιχειά σε αυτή τη προσέγγιση είναι η χρήση πιθανοκρατικών στοιχείων για τη δημιουργία μιας λύσης a priori λύσης αλλά και για την πραγματικού χρόνου (real time) στρατηγική ενημέρωσης της λύσης αυτής.

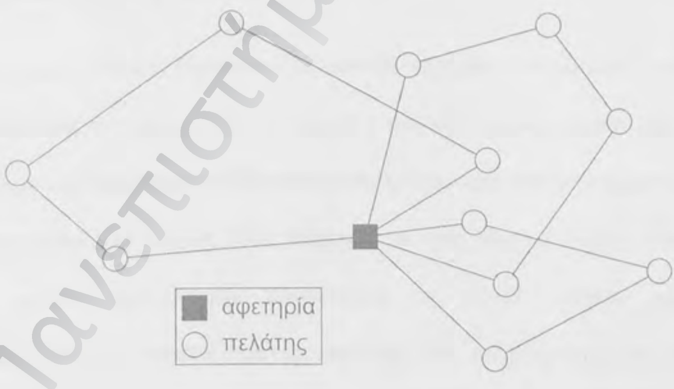
Το Ασύμμετρο Πρόβλημα Περιοδεύοντος Πωλητή με Χρονικά Παράθυρα

Στο Ασύμμετρο Πρόβλημα Περιοδεύοντος Πωλητή με Χρονικά Παράθυρα (A-TSP With Time Windows) τα τόξα του γραφήματος αντιστοιχούν στις μεταβάσεις εργασίας (job transitions), δηλαδή στους χρόνους οργάνωσης (set-up times) που απαιτούνται για να αρχίσει η επεξεργασία μιας εργασίας, η οποία απεικονίζεται ως κόμβος. Για κάθε εργασία, δίνονται από το πρόβλημα οι εξής πληροφορίες: ένας χρόνος επεξεργασίας (processing time) p, ένα χρονικό διάστημα [r,d] μέσα στο οποίο πρέπει να ξεκινήσει η επεξεργασία της εργασίας και το χρονικό αυτό διάστημα ονομάζεται χρονικό παράθυρο (time window). Οι Ascheuer, Fischetti & Grotschel[11], ορίζουν ως σκοπό του προβλήματος την εύρεση της διαδρομής ελάχιστου κόστους που επισκέπτεται όλους τους κόμβους και ικανοποιεί τους περιορισμούς του χρονικού παράθυρου. Άρα πρέπει για κάθε κόμβο του γραφήματος (εργασία) ο χρόνος που ξεκινά η επεξεργασία του να βρίσκεται μέσα στο εκάστοτε χρονικό παράθυρο.

[11] Ascheuer N., Fischetti M. and Grötschel M. ,Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut . Journal of Mathematical Programming Vol 90, Iss 3, (2001) p.475-506.

2.3 Το Πρόβλημα Δρομολόγησης Οχημάτων

Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem) παρουσιάστηκε το 1959 από τους Dantzig και Ramser[12] αλλά ουσιαστικά όπως και με το πρόβλημα του περιοδεύοντος πωλητή πρόκειται για μια οικογένεια προβλημάτων. Σκοπός είναι ο προσδιορισμός των δρομολογίων ενός στόλου οχημάτων που έχει μια ή και περισσότερες αφετηρίες, γύρω από ένα σύνολο πόλεων/πελατών. Δηλαδή πρόθεσή μας είναι η εξυπηρέτηση των απαιτήσεων πελατών/πόλεων και η χάραξη της ελάχιστης διαδρομής ξεκινώντας και επιστρέφοντας στην αφετηρία/υπόστεγο (depot). Το παρακάτω σχήμα δείχνει το πρόβλημα και τη προτεινόμενη λύση του για μια αφετηρία.



Σχήμα 2.1: Παρουσίαση προβλήματος δρομολόγησης

Το Πρόβλημα Δρομολόγησης Οχημάτων είναι NP-hard και η ουσία αυτού του δύσκολου προβλήματος βρίσκεται μεταξύ δύο άλλων γνώστων προβλημάτων, του Προβλήματος του

[12] Dantzig G. B. and Ramser R.H. (1959) "The Truck Dispatching Problem". Management Science 6, 80–91.

Περιοδεύοντος Πωλητή (TSP) και του Προβλήματος Κυτίου Συσκευασίας (Bin Packing Problem).

Εάν η χωρητικότητα C των οχημάτων είναι άπειρη, τότε το πρόβλημα μετατρέπεται στο Πολλαπλό Πρόβλημα του Περιοδεύοντος Πωλητή (mTSP) από το οποίο είναι δυνατό να περάσουμε στο κλασσικό TSP προθέτοντας στο γράφημα $k-1$ (όπου k ο αριθμός των διαδρομών) αντίγραφα του κόμβου 0 (αφετηρία/depot) και των ακμών του, και όταν δεν υπάρχουν ακμές(edges) μεταξύ των k κορυφών-αφετηριών (depots).

Μια εφικτή λύση του Προβλήματος Δρομολόγησης Οχημάτων (VRP) είναι ένα στιγμιότυπο (instance) του Προβλήματος Κυτίου Συσκευασίας (Bin Packing Problem). Το αντίστοιχο πρόβλημα απόφασης είναι ένα VRP όπου το κόστος κάθε ακμής (edge) θεωρείται μηδέν, άρα όλες οι εφικτές λύσεις έχουν το ίδιο κόστος.

Το Πρόβλημα Δρομολόγησης Οχημάτων με αρκετή επιτυχία μοντελοποιεί καταστάσεις στο χώρο των μεταφορών, των εταιριών διανομής, εφοδιαστικής και μεταφορών (logistics). Χρήση τέτοιων μοντέλων και η μαθηματική βελτιστοποίηση τους, αποφέρουν σημαντική μείωση των εξόδων που κυμαίνεται από 5% έως 20%, σύμφωνα με τους Toth και Vigo[13]. Στην πράξη όμως εμφανίζονται πολλοί παράπλευροι περιορισμοί, οι οποίοι γίνεται προσπάθεια να μοντελοποιηθούν από τις τροποποιημένες εκδόσεις του προβλήματος που περιγράφονται παρακάτω.

---

[13] Toth P., Vigo D.(2001): "The Vehicle Routing Problem". Monographs on Discrete Mathematics and Applications. SIAM. Philadelphia.

Μαθηματική Έκφραση του Προβλήματος Δρομολόγησης Οχημάτων

Με το Πρόβλημα Δρομολόγησης Οχημάτων (VRP) μπορεί να εκφραστεί ως ένα γράφημα $G(V,E)$, με $V=\{v_o,v_1,...,v_n\}$ το σύνολο των κορυφών (vertex set) και ως v_o την αφετηρία/υπόστεγο (depot). Έστω:

$V'=V\setminus\{v_0\}$ ένα σύνολο από n πόλεις.

$A=\{v_i,v_j\}$ και $v_i,v_j\in V; i\neq j$ είναι το σύνολο των τόξων.

$C=[c_{ij}]$ πίνακας κόστους/αποστάσεων, c_{ij} κόστος/απόσταση μεταξύ πελατών v_i και v_j

d_{v_j} είναι η ζήτηση του κόμβου/πελάτη v_j

R_i είναι η διαδρομή του οχήματος i

m είναι ο αριθμός των οχημάτων και σε καθένα ανατίθεται μία διαδρομή

Όταν $c_{ij}=c_{ji}$ για κάθε $(v_i,v_j)\in A$ το πρόβλημα καλείται συμμετρικό και το σύνολο A αντικαθιστάται από το σύνολο των ακμών (edge set) $E=\{(v_i,v_j)\mid v_i,v_j\in V; i<j\}$.

Σε κάθε κορυφή v_i στο V' ορίζεται μια ποσότητα (quantity) q_i προϊόντων που πρέπει να παραδοθούν από κάποιο όχημα. Το Πρόβλημα Δρομολόγησης Οχημάτων αποτελείται από τον προσδιορισμό ενός συνόλου m διαδρομών οχημάτων ελάχιστου κόστους, ξεκινώντας από και τερματίζοντας στην αφετηρία/υπόστεγο (depot), έτσι ώστε ένα όχημα να επισκεφθεί ακριβώς μια φορά κάθε κορυφή στο V'. Το κάτω φράγμα του αριθμού των οχημάτων που χρειάζονται για την εξυπηρέτηση των πελατών στο σύνολο V είναι $b(V)=\left\lceil \sum_{v_i \in V} d_{v_i} / C \right\rceil$ όπου C η χωρητικότητα ενός οχήματος.

Συνήθως η έκφραση του προβλήματος περιλαμβάνει και χρονικό περιορισμός όπως παρακάτω. Ο χρόνος εξυπηρέτησης των πελατών (service time) δ_i, είναι ο χρόνος που χρειάζεται ένα όχημα για να ξεφορτώσει τη ποσότητα q_i στο v_i και απαιτείται η συνολική διάρκεια κάθε διαδρομής, δηλαδή ο χρόνος ταξιδίου (travel time) c_{ij} και εξυπηρέτησης πελατών δ_i (service time), να μη ξεπερνά ένα όριο T.

Το ΠΔΟ (VRP) όπως περιγράφεται παραπάνω είναι NP-hard όπως έδειξαν οι Lenstra και Rinnooy[14]. Μια εφικτή λύση του προβλήματος αποτελείται από ένα διαμερισμό $R_1,\ldots,R_m$ του V και μια μετάθεση σ_i του $R_i \cup 0$ ορίζοντας τη σειρά των πελατών στη διαδρομή i.

Το κόστος μίας διαδρομής $R_i=\{v_0, v_1,\ldots,v_{m+1}\}$, όπου $v_i \in V$ και $v_0 = v_{m+1} = 0$ με 0 την αφετηρία (depot) δίδεται από την εξίσωση $C(R_i)=\sum_{i=0}^{m} c_{i,i+1} + \sum_{i=0}^{m} \delta_i$. Μια διαδρομή R_i είναι εφικτή όταν ένα όχημα επισκεφθεί ακριβώς μία φορά κάθε πελάτη και η συνολική διάρκεια της διαδρομής δεν υπερβαίνει το όριο $C(R_i) \leq T$. Η αντικειμενική συνάρτηση του κόστους της λύσης S του προβλήματος είναι $F(S)=\sum_{i=0}^{m} C(R_i)$.

[14] Lenstra J. K., Rinnooy Kan A.H.G.,(1981) "Complexity of vehicle routing and scheduling problems", Networks, vol. 11,issue 2, pp. 221-227. 1981

Τροποποιήσεις του Προβλήματος Δρομολόγησης Οχημάτων

Η αρχική περιγραφή του προβλήματος που δίδεται παραπάνω, αποτέλεσε το γεννήτορα διαφόρων τροποποιημένων εκδοχών.

Πρόβλημα Δρομολόγησης Οχημάτων με περιορισμό Χωρητικότητας

Το Πρόβλημα Δρομολόγησης Οχημάτων με περιορισμό Χωρητικότητας (Capacitated VRP) είναι ουσιαστικά το ίδιο με το κλασσικό VRP *αλλά* με τον πρόσθετο περιορισμό ότι όλα τα οχήματα έχουν την ίδια χωρητικότητα. Συνεπώς του ΠΔΟ με περιορισμό χωρητικότητας είναι ένα συνδυαστικό πρόβλημα όπου ένας ορισμένος αριθμός οχημάτων ίδιας χωρητικότητας πρέπει να εξυπηρετήσουν τις ανάγκες των πελατών για ένα προϊόν από μια κοινή αφετηρία με σκοπό την ελαχιστοποίηση του κόστους.

Σαφέστερα ο σκοπός είναι η ελαχιστοποίηση του στόλου των οχημάτων και του χρόνου των διαδρομών, *αλλά* η συνολική ζήτηση των προϊόντων για κάθε διαδρομή πρέπει να είναι εντός των ορίων χωρητικότητας. Μια διαδρομή είναι εφικτή λύση όταν η συνολική ποσότητα που έχει ανατεθεί σε αυτή διαδρομή δεν υπερβαίνει τη χωρητικότητα του οχήματος που κινείται σε αυτή.

Μαθηματικά το πρόβλημα εκφράζεται αφού ορίσουμε τη μεταβλητή Q που είναι η χωρητικότητα κάθε οχήματος. Η λύση είναι ίδια με αυτή του κλασικού VRP *αλλά* με τον επιπρόσθετο περιορισμό ότι η συνολική ζήτηση όλων των πελατών της διαδρομής R_i δεν υπερβαίνει τη χωρητικότητα του οχήματος, δηλαδή $\sum_{v_i \in I} d_{v_i} \leq Q$.

Πρόβλημα Δρομολόγησης Οχημάτων με Πολλαπλές Αφετηρίες

Πρόβλημα Δρομολόγησης Οχημάτων με Πολλαπλές Αφετηρίες (Multiple Depot VRP) ξεκινά από το γεγονός ότι μια εταιρία μπορεί να έχει αρκετές αφετηρίες/υπόστεγα (depots). Εάν οι πελάτες είναι ομαδοποιημένοι (clustered) γύρω από αφετηρίες το πρόβλημα είναι ίδιο με το κλασικό VRP, εάν όμως δεν υπάρχει σαφής ομαδοποίηση τότε είναι ΠΔΟ με Πολλαπλές Αφετηρίες.

Το πρόβλημα ορίζει ένα στόλο οχημάτων για κάθε αφετηρία/depot που πρέπει να ξεκινήσει και να επιστρέψει σε αυτό, έχοντας εξυπηρετήσει τους πελάτες. Σκοπός είναι η ελαχιστοποίηση του στόλου οχημάτων, των διαδρομών και αυτό να επιτευχθεί από πολλαπλές αφετηρίες. Για να θεωρηθεί μια λύση εφικτή πρέπει για κάθε διαδρομή να ισχύουν οι περιορισμοί του κλασικού προβλήματος και πρέπει ξεκινά και να τερματίζει στην ίδια αφετηρία.

Στη μαθηματική έκφραση του προβλήματος απεικονίζουμε το σύνολο των κορυφών ως $V = \{v_1, v_2, ..., v_n\} \cup V_0$ όπου $V_0 = \{v_{01}, ..., v_{0d}\}$ είναι οι κορυφές που απεικονίζουν τις αφετηρίες και μια διαδρομή i ορίζεται ως $R_i = \{d, v_1, ..., v_m, d\}$ με $d \in V_0$. Το κόστος υπολογίζεται όπως και στο κλασικό πρόβλημα.

Στοχαστικό Πρόβλημα Δρομολόγησης Οχημάτων

Το Στοχαστικό Πρόβλημα Δρομολόγησης Οχημάτων (Stochastic VRP) είναι ένα πρόβλημα δρομολόγησης όπου τουλάχιστον ένα μέρος του προβλήματος είναι τυχαίο. Τέτοια μπορεί να είναι :

Στοχαστικοί πελάτες όπου η ένταξη στο πρόβλημα ενός πελάτη v_i αντιστοιχεί στη πιθανότητα p_i ενώ η μη ένταξη του στη πιθανότητα $1-p_i$.

Στοχαστική ζήτηση όπου η ζήτηση d_i κάθε πελάτη είναι τυχαία μεταβλητή.

Στοχαστικοί χρόνοι όπου οι χρόνοι εξυπηρέτησης δ_i και ταξιδίου t_{ij} είναι τυχαίες μεταβλητές.

Στόχος είναι η ελαχιστοποίηση του στόλου οχημάτων, του χρόνου ταξιδίου που απαιτούνται για την ικανοποίηση των πελατών με τις τυχαίες μεταβλητές σε κάθε εκτέλεση.

Το πρόβλημα προσεγγίζεται σε δύο στάδια. Στο πρώτο στάδιο μια λύση προσδιορίζεται χωρίς να λαμβάνονται υπόψη οι τυχαίες μεταβλητές. Στο δεύτερο στάδιο διορθώνεται η λύση έτσι ώστε να συμπεριλαμβάνει και τις εκάστοτε τυχαίες μεταβλητές.

Η μαθηματική έκφραση του προβλήματος εξαρτάται από το ποιο μέρος του προβλήματος είναι τυχαίο.

Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμό Χρόνου

Το Πρόβλημα Δρομολόγησης Οχημάτων με περιορισμό Χρόνου (VRP with Time Windows) βασίζεται στο κλασικό VRP με τον πρόσθετο περιορισμό ότι κάθε πελάτης πρέπει να εξυπηρετηθεί εντός ενός χρονικού περιθωρίου (time window). Για κάθε πελάτη $v \in V$, θέτουμε το διάστημα $[e_v, l_v]$ στο οποίο πρέπει να εξυπηρετηθεί, με e_v το νωρίτερο και l_v το αργότερο που μπορεί να εξυπηρετηθεί ο πελάτης/κόμβος. Το διάστημα $[e_0, l_0]$ στην αφετηρία/depot καλείται ορίζοντας χρονοπρογραμματισμού (scheduling horizon). Ο Solomon[15] θεωρεί ότι αρχικά όλα τα οχήματα φεύγουν από την αφετηρία το συντομότερο δυνατόν χρόνο e_0.

Σκοπός του προβλήματος είναι η ελαχιστοποίηση του στόλου των οχημάτων, των αποστάσεων και του χρόνου που χρειάζεται για να εξυπηρετηθεί κάθε πελάτης. Μια διαδρομή είναι εφικτή λύση όταν στο κλασικό VRP θεωρήσουμε τους περιορισμούς χρόνου, δηλαδή ότι ο πελάτης (κόμβος) πρέπει να εξυπηρετείται εντός των χρονικών ορίων και κάθε διαδρομή πρέπει να ξεκινά και να τερματίζεται εντός του χρονικού περιθωρίου που σχετίζεται με την αφετηρία./depot. Επίσης αν ένα όχημα φθάσει νωρίτερα στο πελάτη(κόμβο),δηλαδή πριν από το κάτω όριο του χρονικού περιορισμού (time window), αυτό δημιουργεί καθυστέρηση στη διαδρομή.

Μαθηματικά το πρόβλημα εκφράζεται παρακάτω.

Έστω η μεταβλητή b_v που αναπαριστά την έναρξη της εξυπηρέτησης του πελάτη v. Επίσης, για να είναι η διαδρομή $R_i = (v_0, v_1, \dots, v_m, v_{m+1})$ εφικτή λύση πρέπει $e_{v_i} \leq b_{v_i} \leq l_{v_i}$ με $1 \leq i \leq m$ και $b_{v_m} + \delta_{v_m} + c_{v_m,0} \leq l_0$. Εφόσον το όχημα μεταβαίνει αμέσως στον επόμενο πελάτη μόλις έχει ολοκληρώσει την εξυπηρέτηση του πελάτη b_{v_i}, μπορούμε να υπολογίσουμε $b_{v_i} = \max\{e_{v_i}, b_{v_{i-1}} + \delta_{v_{i-1}} + c_{v_{i-1}v_i}\}$ με $b_0 = e_0$ και $\delta_0 = 0$. Άρα ο πελάτης v_i έχει χρόνο αναμονής (waiting time) $w_{v_i} = \max\{e_{v_i}, b_{v_i} - b_{v_{i-1}} - \delta_{v_{i-1}} - c_{i-1,i}\}$. Το συνολικό κόστος της διαδρομής i είναι

$$C(R_i) = \sum_{i=0}^{m} c_{i,i+1} + \sum_{i=0}^{m} \delta_i + \sum_{i=0}^{m} w_{v_i}$$

[15] Solomon M., (1987) Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints, Operations Research 35, p. 254-365.

Για μια λύση S με διαδρομές $R_1, R_2, ..., R_m$, το κόστος της S δίδεται από τη συνάρτηση $F(S) = \sum_{i=1}^{m} C(R_i) + M$, όπου M μια μεγάλη σταθερά.

Η σταθερά M προστίθεται διότι ως κύριος σκοπός του προβλήματος θεωρείται η ελαχιστοποίηση του στόλου των οχημάτων. Η λύση S θεωρείται εφικτή όταν όλες οι διαδρομές της είναι εφικτές και κάθε πελάτης ανήκει σε ακριβώς μια διαδρομή.

2.4 Τοποθέτηση Κατασκευαστικών Κυψελών ως Πρόβλημα Συνδυαστικής Βελτιστοποίησης

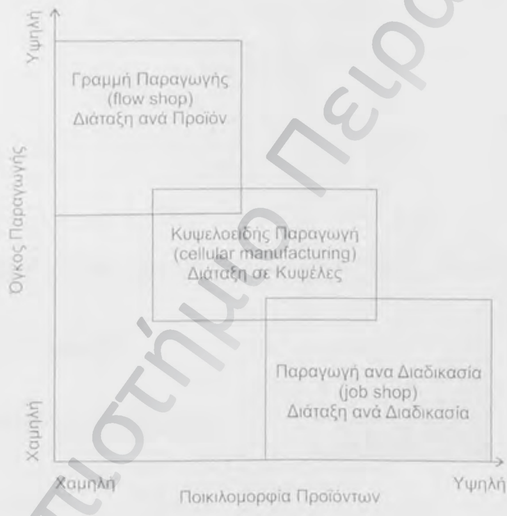
Ένας από τους πλέον σημαντικούς παράγοντες για το σχεδιασμό εργοστασίων είναι η εύρεση μια αποτελεσματικής τοποθέτησης στο χώρο των μηχανών. Ως αποτελεσματική ορίζουμε τη διάταξη στο χώρο που έχει υψηλή συνάφεια με τις διαδικασίες παραγωγής. Από την τοποθέτηση των μηχανών επηρεάζεται το κόστος παραγωγής καθώς άσκοπες μετακινήσεις των πρώτων υλών εντός του εργοστασίου από μηχανή σε μηχανή μειώνουν την παραγωγικότητα και την απόδοση του εργοστασίου.

Μέθοδοι Παραγωγής και Είδη Τοποθέτησης

Συνήθως η τοποθέτηση των μηχανών εξαρτάται από το είδος της παραγωγικής διαδικασίας[16]. Στη περίπτωση που ο όγκος παραγωγής είναι υψηλός και η ποικιλομορφία των προϊόντων μικρή, η παραγωγική διαδικασία χαρακτηρίζεται ως πρόβλημα ροής (flow shop process)και η

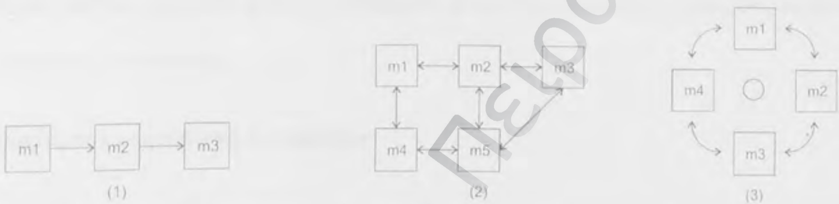
[16] Dilworth J.B., Operation Management, 2nd Edition, McGraw Hill, 1996

τοποθέτηση βασίζεται στα προϊόντα. Από αυτό η τοποθέτηση καλείται *ανα προϊόν* (layout by product). Η αντίθετη μορφή παραγωγικής διαδικασίας έχει υψηλή ποικιλομορφία προϊόντων και σχετικά μικρό όγκο παραγωγής. Εκεί η διαδικασία χαρακτηρίζεται ως παραγωγή ανά εργασία και αντίστοιχα η τοποθέτηση των μηχανών *ανά εργασία* ή *ανά διαδικασία* (job shop layout, layout by process). Η κυψελοειδής διάταξη παραγωγής (cellular manufacturing) εφαρμόζεται σε παραγωγικές διαδικασίες ανάμεσα σε στις δύο προαναφερθείσες όπως φαίνεται και στο σχήμα 1.



Σχήμα 2.2. Οι διαφορετικές διατάξεις παραγωγής σε σχέση με τη ποικιλομορφία και τον όγκο παραγωγής των προϊόντων. Προσαρμογή από Dilworth 1996.

Στη κυψελοειδή παραγωγή το πρόβλημα της διάταξης της των μηχανών (machine layout problem) είναι η εύρεση της καλύτερης τοποθέτησής τους στο χώρο καθώς οι πιο διαδεδομένες τοποθετήσεις είναι τρείς. Σε μονή γραμμή, σε πολλαπλές γραμμές και σε διάταξη βρόχου. Στη διάταξη μονής γραμμής τα προϊόντα που παράγονται χρησιμοποιούν την ίδια ακολουθία μηχανών (machine sequence) και το προϊόν κινείται σε μια γραμμή ροής (flow line). Στη διάταξη σε πολλαπλές γραμμές υπάρχει η δυνατότητα κίνησης προς οποιαδήποτε μηχανή της κυψέλης και η ροή μπορεί να κατευθυνθεί προς οποιαδήποτε κατεύθυνση. Στη διάταξη βρόχου οι μηχανές τοποθετούνται σε σχεδόν κυκλικά και η κίνηση των προϊόντων γίνεται και προς τις δύο κατευθύνσεις της περιφέρειας του κύκλου. Το σχήμα 3 δείχνει τις προαναφερθείσες κινήσεις και διατάξεις. Στη γραμμική διάταξη γίνεται προσπάθεια να σχετίζεται σε υψηλό βαθμό η τοποθέτηση των μηχανών με όσο το δυνατόν περισσότερα εξαρτήματα/προϊόντα.



Σχήμα 2.3. Ροές μεταξύ μηχανών σε διατάξεις παραγωγής.

(1) Διάταξη μονής γραμμής

(2) Διάταξη σε πολλαπλές γραμμές

(3) Διάταξη βρόχου

Απαιτούμενα Δεδομένα για την Ανάπτυξη Κυψέλης Παραγωγής

Η ανάπτυξη μιας κυψέλης παραγωγής περιλαμβάνει τρία βήματα.

- Δημιουργία μια οικογένειας εξαρτημάτων/προϊόντων που πρέπει να παραχθούν
- Τοποθέτηση μηχανών εντός της κυψέλης
- Τοποθέτηση κυψελών με βάση τις μεταξύ τους σχέσεις

Τα αρχικά δεδομένα που απαιτούνται είναι :

* Πίνακας Μηχανών/Εξαρτημάτων
* Ακολουθία των μηχανών που χρησιμοποιούνται για την παραγωγή κάθε εξαρτήματος

Τα δεδομένα της ακολουθίας των μηχανών χρησιμοποιούνται για την επιλογή των μηχανών που θα ενταχθούν στη κυψέλη και για την τοποθέτηση τους εντός της. Επίσης με αυτά τα δεδομένα δημιουργείται ένα γράφημα ροών εξαρτημάτων/προϊόντων από και προς για κάθε μηχανή. Η έκταση που χρειάζεται η κάθε μηχανή χρησιμοποιείται για να εκτιμηθεί το μέγεθος της κυψέλης και η απόσταση μεταξύ τους.

Παράγοντες που επηρεάζουν το σύστημα

Σημαντικοί παράγοντες στο προσδιορισμό του προβλήματος είναι ο αριθμός και ο τύπος των κινήσεων μεταξύ των μηχανών καθώς και ο τρόπος χειρισμού των εξαρτημάτων/προϊόντων.

Οι δύο πιο σημαντικές κινήσεις σε μια γραμμή ροής είναι προς τα πίσω ροή (backward flow) και σε παράκαμψη (bypassing flow). Η προς τα πίσω ροή συμβαίνει όταν ένα εξάρτημα/προϊόν πρέπει να μετακινηθεί από τη μηχανή που βρίσκεται σε μια άλλη προηγούμενη μηχανή στη γραμμή παραγωγής. Αυτό μπορεί να απαιτείται πολλές φορές, καθώς ορίζεται από την ακολουθία παραγωγής του κάθε εξαρτήματος/προϊόντος ξεχωριστά. Η κίνηση σε παράκαμψη συμβαίνει όταν ένα εξάρτημα/προϊόν πρέπει να μετακινηθεί από τη μηχανή που βρίσκεται σε μια επόμενη μηχανή η οποία δεν είναι η ακριβώς επόμενη μηχανή στη γραμμή παραγωγής. Αυτοί οι δύο τύποι κινήσεων θεωρούνται οι πλέον ανεπιθύμητοι καθώς διακόπτουν τη ροή και συνεπώς επηρεάζουν αρνητικά την παραγωγικότητα.

Μαθηματική Προσέγγιση της Βέλτιστης Τοποθέτησης Μηχανών και το Πρόβλημα της Τετραγωνικής Ανάθεσης

Το Πρόβλημα της Τετραγωνικής Ανάθεσης

Το Πρόβλημα της Τετραγωνικής Ανάθεσης (Quadratic Assignment Problem) έχει εισαχθεί από τους Koopmans και Beckmann[17] το 1957. Περιγράφεται ως : Όταν θέσουμε n εγκαταστάσεις (facilities) και n τοποθεσίες (locations), και δύο πινάκες $n \times n$, $A=[a_{ij}]$ και $B=[b_{rs}]$, όπου a_{ij} η απόσταση μεταξύ τοποθεσίας i και j και b_{rs} η ροή μεταξύ των εγκαταστάσεων r και s, ζητείται η μετάθεση π^* που ελαχιστοποιεί την παρακάτω συνάρτηση

$$\min_{\pi \in \Pi(n)} f(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij} \cdot a_{\pi_i \pi_j} \qquad (1)$$

όπου $\Pi(n)$ είναι το σύνολο των μεταθέσεων και π_i η τοποθεσία της εγκατάστασης i στη τρέχουσα λύση $\pi \in \Pi(n)$. Ο όρος $b_{ij} \cdot a_{\pi_i \pi_j}$ δείχνει τη συνεισφορά στο κόστος της συνάρτησης όταν ταυτόχρονα αναθέτουμε την εγκατάσταση i στη τοποθεσία π_i και την εγκατάσταση j στη τοποθεσία π_j. Παρόλη τη κομψή και σχετικά απλή του εξίσωση ως γνήσιο πρόβλημα συνδυαστικής βελτιστοποίησης έχει δειχθεί ως NP-Hard από τους Shani και Gonzalez[18] και θεωρείται από τα πλέον δύσκολα .

[17] Koopmans TC, Beckman M. Assignment problems and the location of economic activities. Econometrica, Vol. 25, No. 1 (Jan., 1957), pp. 53-76

[18] Shani S and Gonzalez T (1976). P-complete approximation problems. Journal of the ACM **23**: 555–565.

Το Πρόβλημα της Τετραγωνικής Ανάθεσης και Κατασκευαστικές Διαδικασίες

Το Τετραγωνικό Πρόβλημα Ανάθεσης (Quadratic Assignment Problem) ορίζεται ως δύο πίνακες n επί n : τον πίνακα ροής F του οποίου το στοιχείο (i,j) αναπαριστά τη ροή μεταξύ της δραστηριότητας (facility) i και της δραστηριότητας j, και τον πίνακα D του οποίου το στοιχείο (i,j) αναπαριστά την απόσταση μεταξύ της τοποθεσίας (location) i και της τοποθεσίας j. Αναπαριστούμε την ανάθεση με το άνυσμα π το οποίο είναι μια μετάθεση των αριθμών $\{1,2,....,n\}$.

Σαφέστερα το πρόβλημα μπορεί να οριστεί ως εξής: εξετάζουμε ένα σύνολο δραστηριοτήτων n, οι οποίες πρέπει να ανατεθούν σε n θέσεις (ή αντίστροφα).

Μια μήτρα $D=[d_{ij}]_{n,n}$ δίνει τις αποστάσεις μεταξύ των θέσεων (location), όπου d_{ij} είναι η απόσταση μεταξύ της θέσης i και της θέσης j. Μια μήτρα $F=[f_{hk}]_{n,n}$ χαρακτηρίζει τις ροές μεταξύ των δραστηριοτήτων (μεταφορές στοιχείων, του υλικού, των ανθρώπων κ.τ.λ.) όπου f_{hk} είναι η ροή μεταξύ της δραστηριότητας h και της δραστηριότητας k.

Μια ανάθεση (assignment) είναι μια μετάθεση π {1,.....n} όπου $\pi(i)$ είναι η δραστηριότητα που ορίζεται στην θέση i. Το πρόβλημα είναι να βρεθεί μια μετάθεση $\pi(n)$ έτσι ώστε το προϊόν των ροών μεταξύ των δραστηριοτήτων να ελαχιστοποιείται από τις αποστάσεις μεταξύ των θέσεων τους. Τυπικά το QAP μπορεί να διατυπωθεί ως πρόβλημα μετάθεσης 'π' που ελαχιστοποιεί την ακόλουθη συνάρτηση:

$$c(\pi) = \min_{\pi \in \Pi} \sum_{i,j=1}^{n} d_{ij} f_{\pi(i)\pi(j)}$$

Ο Solimanpur[19] τροποποιεί το πρόβλημα της τετραγωνικής ανάθεσης με σκοπό να το προσαρμόσει στις κυψελοειδείς διαδικασίες παραγωγής. Από την υπόθεση ότι η σχετική απόσταση μεταξύ κάθε ζεύγους μηχανών είναι γνωστή το πρόβλημα εκφράζεται μαθηματικά από την παρακάτω εξίσωση και τους περιορισμούς της.

$$Min\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{l=1}^{M}\sum_{h=1}^{M} f_{ij}\cdot d_{lh}\cdot x_{il}\cdot x_{jh} \qquad (2)$$

με $$\sum_{l=1}^{M} x_{il}=1, \quad i=1,2,...,M \qquad (3)$$

$$\sum_{i=1}^{M} x_{il}=1, \quad i=1,2,...,M \qquad (4)$$

$$x_{il}\in\{0,1\} \quad i,l\in\{1,2,...,M\} \qquad (5)$$

και f_{ij} :η συχνότητα ροής από τη μηχανή i στη μηχανή j

d_{ij} :η απόσταση μεταξύ της τοποθεσίας i και της τοποθεσίας j

$$x_{il}=\begin{cases}1, & \text{εάν η μηχανή i ανατίθεται στη τοποθεσία l} \\ 0, & \text{αλλιώς}\end{cases};$$

M : το συνολικό πλήθος των μηχανών και τοποθεσιών

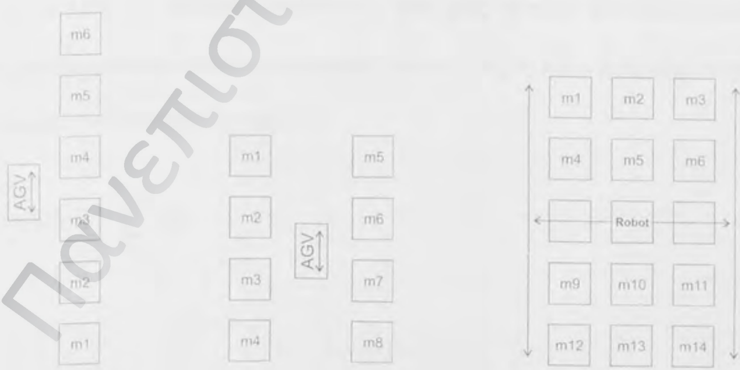
Η αντικειμενική συνάρτηση (2) ελαχιστοποιεί το σύνολο των αποστάσεων των υλικών ροών. Ο περιορισμός που τίθεται με την εξίσωση (3) δηλώνει ότι σε κάθε τοποθεσία αντιστοιχεί μονό μία μηχανή. Ο περιορισμός που τίθεται με την εξίσωση (4) δηλώνει ότι σε κάθε μηχανή αντιστοιχεί

[19] Solimanpur M., Vrat P., Shankar R., (2004). Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing. European Journal of Operational Research 157 :592-606

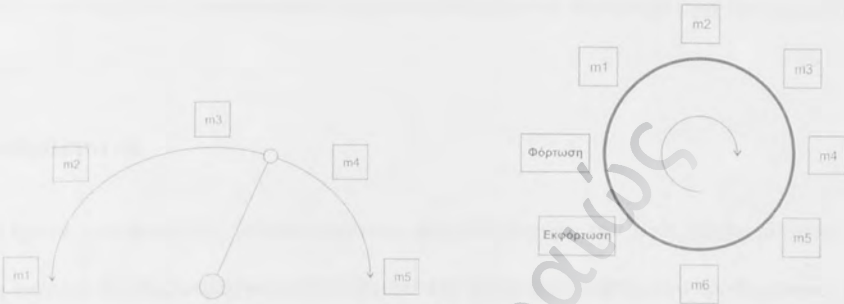
μονό μία τοποθεσία. Η εξίσωση (5) αναπαριστά τις δυαδικές μεταβλητές. Μια πλήρης λύση του προβλήματος αναπαρίσταται από μια πλήρη μετάθεση των κατασκευαστικών μηχανών που δηλώνονται ως π. Έστω η μετάθεση $\pi=(3,2,1,4)$, μια εφικτή και πλήρης λύση ενός προβλήματος τοποθέτησης μηχανών με τέσσερις μηχανές. Οι μηχανές 3,2,1 και 4 ανατίθενται αντίστοιχα στις τοποθεσίες 1,2,3 και 4.

Η Διάταξη Μονής Σειράς

Η τοποθέτηση μιας σειράς Μηχανών (single-row machine layout) ή γραμμική τοποθέτηση μηχανών (Linear Machine Layout) ορίζεται ως η τοποθέτηση κατά την οποία οι μηχανές που επεξεργάζονται τα υλικά είναι η μια δίπλα στην άλλη και ο τρόπος μετακίνησης των υλικών μεταξύ τους γίνεται σε μια νοητή ευθεία. Η μετακίνηση των υλικών σε τέτοιου είδους διατάξεις γίνεται συνήθως με ένα σύστημα ιμάντα (conveyor belt) ή με Αυτόματα Καθοδηγούμενο Όχημα (Automated Guided Vehicle).



Σχήμα 2.4. Διατάξεις μηχανών που εξυπηρετούνται από ένα Αυτόματα Καθοδηγούμενο Όχημα: τοποθέτηση μίας γραμμής, διπλής γραμμής με AGV και τοποθέτηση συστοιχίας με ρομποτικό βραχίονα τύπου gantry.



Σχήμα 2.5. Διατάξεις μηχανών που εξυπηρετούνται από ένα Αυτόματα Καθοδηγούμενο Όχημα : Κυκλική τοποθέτηση με ένα ρομποτικό βραχίονα και τοποθέτηση κλειστού βρόγχου με ιμάντα μετακινήσεων

Στη παραγωγή δυστυχώς υπάρχουν προϊόντα που για να κατασκευαστούν δε θα περάσουν με ροή μόνο προς τα εμπρός σε ακολουθία. Ορισμένα πρέπει να μετακινηθούν προς τα πίσω από τον ιμάντα ή το AGV και συνεπώς δημιουργούν ροή προς τα πίσω που είναι ανεπιθύμητη. Σε αυτή την περίπτωση σκοπός είναι η ελαχιστοποίηση της προς τα πίσω ροής (backward flow) και παρουσιάζεται στη συνάρτηση παρακάτω.

$$Obj = \min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} d_{ij}$$

όπου:

f_{ij} :ένας αριθμός που δίνει τη συχνότητα/όγκο των εξαρτημάτων που μετακινούνται

d_{ij} :η απόσταση μεταξύ της μηχανής i και της μηχανής j

Ο πίνακας f_{ij} δείχνει τις κινήσεις από και προς την κάθε μια μηχανή και λαμβάνεται από τις πληροφορίες που ορίζουν την ακολουθία μηχανών που πρέπει να υλοποιηθεί για την παραγωγή κάθε αγαθού.

2.5 Συμπεράσματα

Τα προβλήματα συνδυαστικής βελτιστοποίησης ασχολούνται με την αναζήτηση μεγίστης ή ελάχιστης από ένα πλήθος επιλογών που υπάρχουν στο χώρο αναζήτησης του προβλήματος. Τα προβλήματα βελτιστοποίησης μπορούν να εκφραστούν ως προβλήματα απόφασης αν θέσουμε κάποιον όρο και συνήθως εάν υπάρχει ένας γρήγορος αλγόριθμος για ένα πρόβλημα απόφασης, είναι σχετικά εύκολο να λυθεί και το αντίστοιχο πρόβλημα βελτιστοποίησης. Όμως τέτοιου είδους προβλήματα γίνονται εξαιρετικά δύσκολα όταν συμβαίνει η συνδυαστική έκρηξη.

Η συνδυαστική έκρηξη είναι η εκθετική αύξηση του μεγέθους του χώρου αναζήτησης του προβλήματος και είναι και ο κύριος λόγος που αυτά τα προβλήματα με τη συνήθως μικρή και κομψή μαθηματική έκφραση είναι τόσο δύσκολα να λυθούν και συχνά κατατάσσονται στη κλάση NP (Non deterministic Polynomial). Αυτό σημαίνει ότι τέτοιου είδους προβλήματα μπορούν να εξακριβωθούν (verified) αλλά όχι να λυθούν σε πολυωνυμικό χρόνο. Συνεπώς η κλάση NP είναι η κλάση προβλημάτων απόφασης για την οποία είναι εύκολο να *ελέγξουμε* (check) την ορθότητα της υποτιθέμενης λύσης, με τη βοήθεια της πληροφορίας που προέρχεται από το πιστοποιητικό. Δηλαδή δε ζητάμε τρόπο *εύρεσης* της λύσης αλλά μόνο τρόπο εξακρίβωσης (verify) ότι η υποτιθέμενη λύση είναι πράγματι αληθείς. Οι παραδοσιακές μαθηματικές μέθοδοι προσφέρουν τη εξασφάλιση της απόλυτα βέλτιστης λύσης αλλά όταν το μέγεθος του προβλήματος μεγαλώνει, η συνδυαστική έκρηξη στο χώρο αναζήτησης λύσεων οδηγεί αυτού του είδους τους αλγορίθμους να έχουν απαιτήσεις υπολογιστικού χρόνου που δεν είναι πρακτικά εφαρμόσιμες. Επιπρόσθετα, οι παραδοσιακές μέθοδοι χρειάζονται πολλές φορές περιορισμούς που απλοποιούν τόσο το πρόβλημα ώστε πλέον δεν ανταποκρίνονται στη ουσία του πραγματικού προβλήματος.

Για παράδειγμα, στο πρόβλημα του περιοδεύοντος πωλητή ένα πιστοποιητικό είναι μια διαδρομή (tour) της οποίας το συνολικό μήκος είναι $\leq K$. Ο αλγόριθμος Α θα εξακριβώσει (verify) ότι πράγματι η διαδρομή επισκέπτεται όλες τις πόλεις και πράγματι έχει συνολικό μήκος $\leq K$.

Το πρόβλημα του περιοδεύοντος πωλητή έχει απασχολήσει πλήθος ερευνητών λόγο της απλής έκφρασης του αλλά και της μεγάλης εφαρμογής του, όπως και το σχετικά συναφές από εννοιολογική άποψη πρόβλημα δρομολόγησης οχημάτων . Επίσης το πρόβλημα τετραγωνικής ανάθεσης και η σχέση του με τις διαδικασίες παραγωγής έχουν προσελκύσει πολλές ερευνητικές προσπάθειες. Πραγματικά προβλήματα με πλήθος εφαρμογών μπορούν να αναχθούν σε αυτά τα προβλήματα της κλάσης NP Complete και οι τροποποιήσεις τους με σκοπό να καλύψουν μεγαλύτερο εύρος εφαρμογών δείχνουν το ενδιαφέρον που έχουν προκαλέσει στη ερευνητική κοινότητα καθώς και την αντίστοιχη προσπάθεια που έχει γίνει για την επίλυση τους. Μέχρι να υπάρξει σαφής απόδειξη για το αν η κλάση P είναι ίση ή διάφορη της NP , τα προβλήματα αυτά θα παραμένουν ουσιαστικά δίχως ακριβή λύση και θα συνεχίζουμε να τα αντιμετωπίζουμε με προσεγγιστικούς αλγορίθμους.

Κεφάλαιο 3

Η Κοινωνική Συμπεριφορά των Εντόμων

και η Βιολογική Προέλευση των Ιδεών

Εισαγωγή 3.1

Τα έντομα αποτελούν ίσως την πιο διαδεδομένη μορφή ζωής στο πλανήτη. Αν και μόνο το 2% των εντόμων είναι κοινωνικά, αντιπροσωπεύουν το 50% της βιομάζας παγκοσμίως[1]. Ως κοινωνικά εννοούμε τα έντομα που ζούνε σε αποικίες, δηλαδή όλα τα είδη μυρμηγκιών και τερμιτών, καθώς και ορισμένα είδη μελισσών. Τα μυρμήγκια έχουν αποικίσει κάθε μέρος του πλανήτη μας και αποτελούν μία από τις πιο οργανωμένες μορφές ζωής. Αλλά σε μία αποικία κοινωνικών εντόμων, ένας μέλος, τις περισσότερες φορές, δεν εκτελεί όλων των ειδών εργασίες, αλλά, μάλλον, ειδικεύεται σε ένα σύνολο εργασιών, 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[2].

Κάθε έντομο έχει καθορισμένη δομή και ρόλο ενώ η συμμετοχή του στην αποικία συμβάλει στην οργανωμένη μορφή της καθώς η ολοκλήρωση όλων των δραστηριοτήτων των εντόμων δεν απαιτεί την ύπαρξη «αρχηγού».

[1] Arnett R.H., American Insects: A handbook of the insects of America north of Mexico. Van Nostrand Rehinhold, New York, 1985

[2] Jeanne, R. L., (1986). The Evolution of the Organization of Work in Social Insects, *Monit. Zool. Ital.*, 20, 119-133.

Αυτή η αποκεντρωμένη οργάνωση τους δίνει ως σύνολο την ικανότητα να επιλύουν δύσκολα προβλήματα βελτιστοποίησης που ως μεμονωμένα άτομα δεν θα μπορούσαν. Ως παράδειγμα τέτοιων δραστηριοτήτων προτάσσονται η αναζήτηση της μικρότερης διαδρομής έως την τροφή, ο καταμερισμός εργασιών και η ομαδοποίηση ειδικευμένων μυρμηγκιών. Δραστηριότητες που έχουν αντιστοιχίες και σε ανθρώπινες δραστηριότητες, αν και το ανθρώπινο είδος έχει ένα εξελικτικό 'μειονέκτημα' καθώς οι ανθρώπινες αποικίες υπάρχουν για 50.000 χρόνια έναντι 100 εκατομμυρίων ετών των αποικιών εντόμων.

3.2 Ευφυΐα Σμήνους

Η Ευφυΐα Σμήνους (Swarm Intelligence) είναι τεχνητή ευφυΐα που βασίζεται στη συλλογική συμπεριφορά αποκεντρωμένων, αυτο-οργανομένων συστημάτων. Ο όρος προτάθηκε το 1989 από τον Beni[3] στα πλαίσια των κυψελοειδών ρομποτικών συστημάτων. Ο πληθυσμός των ατόμων του συστήματος αποτελείται από άτομα που είτε είναι όμοια είτε διαφέρουν ελάχιστα μεταξύ τους αλλά είναι σχετικά απλά σαν μονάδες. Αυτό ο πληθυσμός αλληλεπιδρά δίχως να υπάρχει ένα γενικό πλαίσιο κανόνων ή συνειδητή συμπεριφορά των ατόμων και το σύνολο των αλληλεπιδράσεων εκφράζει τη συλλογική ευφυΐα του συστήματος. Αυτό είναι απόρροια της τοπικής αλληλεπίδρασης ατόμων που γειτνιάζουν στο χώρο και της συμπεριφοράς τους που ορίζεται από απλούς κανόνες. Συχνά αυτή η συμπεριφορά του κάθε ατόμου περιγράφεται με στοχαστικούς όρους που βασίζονται στη τοπική αντίληψη του ατόμου για το περιβάλλον του.

3 Beni, G., Wang, J. (1989) Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30.

Η Ε.Σ. περιλαμβάνει φυσικά και τεχνητά συστήματα που αποτελούνται από πολλά άτομα που συντονίζονται με αποκεντρωμένο έλεγχο και αυτο-οργάνωση. Συγκεκριμένα εστιάζει στις συλλογικές συμπεριφορές που αναδύονται (emerge) από τις τοπικές αλληλεπιδράσεις των ατόμων μεταξύ τους και με το περιβάλλον. Παραδείγματα τέτοιων συστημάτων είναι οι αποικίες μυρμηγκιών και τερμιτών, σμήνη πουλιών, κοπάδια ψαριών, αγέλες ζώων και αποικίες μελισσών. Ανθρώπινες εφαρμογές με τέτοιες συμπεριφορές είναι πολύ-ρομποτικά συστήματα και λογισμικό με σκοπό τη βελτιστοποίηση.

Ιδιότητες Συστημάτων Ευφυΐας Σμήνους

Το τυπικό σύστημα Ευφυΐας Σμήνους έχει τις ακόλουθες ιδιότητες:

- Αποτελείται από πλήθος ατόμων
- Τα άτομα είναι σχετικά ομοιογενή, δηλαδή είτε είναι όμοια είτε διαφέρουν ελάχιστα
- Οι αλληλεπιδράσεις μεταξύ των ατόμων βασίζονται σε απλούς κανόνες συμπεριφοράς που βασίζονται μόνο σε τοπικές πληροφορίες που τα άτομα αποκτούν είτε απευθείας είτε από το περιβάλλον
- Η συλλογική συμπεριφορά του συστήματος είναι απόρροια των αλληλεπιδράσεων των ατόμων μεταξύ τους και με το περιβάλλον τους, δηλαδή η συμπεριφορά του συστήματος αυτο-οργανώνεται.

Η χαρακτηριστική ιδιότητα ενός συστήματος ευφυΐας σμήνους είναι η δράση με συντονισμένο τρόπο δίχως τη παρουσία ενός κεντρικού μηχανισμού συντονισμού ή εξωτερικού ελέγχου.

3.3 Αυτο-Οργάνωση

Αποκεντρωμένη οργάνωση όμως δε σημαίνει έλλειψη οργάνωσης. Το κρίσιμο στοιχείο για την επιτυχία των συλλογικών εργασιών της αποικίας είναι η αυτό-οργάνωση (self-organization). Ο όρος αυτός περιγράφει τις πολύπλοκες συμπεριφορές που αναδύονται (emerge) από την αλληλεπίδραση των σχετικά απλών ατόμων-πρακτόρων. Στο μοντέλο της αυτό-οργάνωσης δίνεται έμφαση στο ότι το άτομο-πράκτορας περατώνει μόνο απλές λειτουργίες. Δεν λαμβάνεται υπόψη στο μοντέλο το πόσο πολύπλοκη είναι η δομή του ατόμου για να υλοποιήσει τις δράσεις. Για παράδειγμα η μετακίνηση ενός μυρμηγκιού μεταξύ της φωλιάς και της τροφής απεικονίζεται ως ένα μονοπάτι φερομόνης (pheromone trail). Ένα μυρμήγκι καθώς κινείται στο χώρο εναποθέτει στο έδαφος τη χημική ουσία φερομόνη, η οποία γίνεται αντιληπτή από τα αισθητήρια όργανα των άλλων μυρμηγκιών. Το μοντέλο δεν ασχολείται με το πώς λειτουργεί το αισθητήριο όργανο των μυρμηγκιών αλλά με την απόφαση που λαμβάνουν να ακολουθήσουν ή όχι ένα μονοπάτι φερομόνης, και βάση ποιού κανόνα ενισχύουν τη φερομόνη.

Οι απαρχές των μοντέλων αυτό-οργάνωσης εντοπίζονται στα πεδία της φυσικής και της χημείας[4] όπου απεικονίζουν μικροσκοπικές λειτουργίες που δημιουργούν μακροπρόθεσμα πρότυπα (patterns).

Ο Camazine[5] δίνει τον παρακάτω ορισμό για την αυτο-οργάνωση:

[4] Nicolis G., Prigogine I., Self-organization in Non-Equilibrium Systems, Willey & Sons, 1977
[5] 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

'Η αυτο-οργάνωση είναι μια διαδικασία μέσα στην οποία ένα μοντέλο (model) ολικού επιπέδου αναδύεται (emerges) 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.'

Εξετάζοντας την αυτο-οργάνωση των κοινωνικών εντόμων μιας αποικίας μπορούμε να εξάγουμε χρήσιμα συμπεράσματα για το σχεδιασμό ευφυών συστημάτων. Η αποικία δρα ως ένα αποκεντρωμένο σύστημα επίλυσης προβλημάτων τα οποία επιλύονται από την αλληλεπίδραση απλών πρακτόρων. Η προσέγγιση αυτή έχει ως εγγενή χαρακτηριστικά την ευλιξία (flexibility) και την ευρωστία (robustness). Η ευλιξία οδηγεί την προσαρμογή στις εκάστοτε περιβαλλοντικές συνθήκες, ενώ η ευρωστία επιτρέπει στην αποικία τη δυνατότητα να λειτουργήσει ακόμα και αν κάποια άτομα αποτύχουν.

Η αυτο-οργάνωση των κοινωνικών εντόμων αποτελείται από τεσσάρα βασικά στοιχεία.

- 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

Θετική ανάδραση είναι ένας μηχανισμός βάση του οποίου τα κοινωνικά έντομα συγκλίνουν προς την κατεύθυνση καλών λύσεων ενός προβλήματος. Η κατασκευή δομών με τη θετική ανάδραση που μπορούν να χρησιμοποιηθούν από την αποικία επιτυγχάνεται μέσω της *στρατολόγησης* (recruitment) και της *ενίσχυσης* (reinforcement).

Σαν παράδειγμα *στρατολόγησης* θεωρούμε ένα μυρμήγκι που στο χώρο αναζήτησης, εντοπίζει μια πηγή τροφής και επιστρέφει στη φωλιά. Κατά τη διάρκεια της επιστροφής του στη φωλιά του, εναποθέτει ποσότητες μιας χημικής ουσίας που ονομάζεται φερομόνη, ανάλογες προς την ποσότητα και την ποιότητα της τροφής, στη διαδρομή που ακολουθεί. Η διαδρομή που ακολούθησε αυτό το μυρμήγκι είναι τώρα εμποτισμένη με φερομόνη και ονομάζεται ίχνος ή μονοπάτι φερομόνης (pheromone trail). Το μονοπάτι φερομόνης είναι αντιληπτό από τα υπόλοιπα μυρμήγκια της αποικίας τα οποία είναι πιο πιθανό να επιλέξουν την συγκεκριμένη διαδρομή με το ίχνος φερομόνης από μια άλλη τυχαία διαδρομή. Συνεπώς το πρώτο μυρμήγκι 'στρατολογεί' τα υπόλοιπα και έτσι όλη η αποικία σιγά σιγά στρατολογείται στην εκμετάλλευση αυτής της πηγής τροφής.

Η *ενίσχυση* είναι μηχανισμός που δρα κατά την σύγκλιση προς τη πηγή της τροφής. Καθώς η αποικία τείνει να συγκλίνει προς τη πηγή τροφής, το τυχαίο μυρμήγκι πρέπει να αποφασίσει με κάποια πιθανότητα αν θα ακολουθήσει το μονοπάτι-ίχνος φερομόνης ή όχι. Αν το ακολουθήσει ενισχύει με τη φερομόνη του τη συγκεκριμένη διαδρομή που οδηγεί στη πηγή τροφής την. Αυτή η πρόσθετη φερομόνη θα κάνει τη διαδρομή ακόμα πιο πιθανή να επιλεγεί από τα επόμενα μυρμήγκια.

Ο ρόλος της ενίσχυσης γίνεται πιο σαφείς στο ιδεατό μοντέλο αποικίας μυρμηγκιών (Ant Colony Optimization) και συγκεκριμένα στις επαναλήψεις του εκάστοτε αλγορίθμου.

Αρνητική Ανάδραση

Η αρνητική ανάδραση έχει σαν σκοπό να ισορροπήσει τα αποτελέσματα της θετικής ανάδρασης και να απεικονίσει καταστάσεις όπως την εγκατάλειψη κάποιων διαδρομών, την ύπαρξη δύο ή περισσοτέρων πηγών τροφής ή την εξάντληση κάποιας πηγής τροφής. Έχει παρατηρηθεί ότι στη φύση η φερομόνη που εναποθέτουν τα μυρμήγκια έχει ένα ρυθμό εξάτμισης και αυτό είναι ένα πρακτικό παράδειγμα αρνητικής ανάδρασης.

Ενίσχυση των Τυχαίων Διακυμάνσεων

Καθώς ελλοχεύει ο κίνδυνος να παρασυρθεί η αποικία σε μια τοπικά βέλτιστη λύση η ενίσχυση των τυχαίων διακυμάνσεων είναι ένας μηχανισμός που σκοπεύει να βοηθήσει στην ανακάλυψη(exploration) νέων τοπικών ή ολικών βέλτιστων λύσεων. Η ισορροπία μεταξύ εκμετάλλευσης (explotation) και ανακάλυψης (exploration) λύσεων-πηγών τροφής είναι κρίσιμη για την επιτυχία του συστήματος. Αν για παράδειγμα ένα μυρμήγκι δεν επιλέξει την 'πεπατημένη' διαδρομή φερομόνης υπάρχει πάντα η πιθανότητα να ανακαλύψει μια νέα καλύτερη πηγή τροφής και μέσω της θετικής ανάδρασης να προσφέρει στην αποικία τη νέα πηγή. Βέβαια υπάρχει και η πιθανότητα να χαθεί όπου αφαιρεί ένα άτομο από το σύνολο της αποικίας.

Ύπαρξη Αλληλεπιδράσεων Μεταξύ των Ατόμων της Αποικίας

Οι Πολλαπλές Αλληλεπιδράσεις (multiple interactions) μεταξύ των ατόμων μιας αποικίας είναι θεμελιώδεις για την ύπαρξη ενός μοντέλου αυτο-οργάνωσης. Ένα μονοπάτι-ίχνος φερομόνης (pheromone trail) είναι μια αυτο-οργανωμένη δομή η οποία έχει προκύψει από αλληλεπίδραση των ατόμων της αποικίας. Για να συμβεί αυτό όμως απαιτείται ένας ελάχιστος αριθμός ατόμων στην αποικία. Πράγματι έχει παρατηρηθεί ότι μια αποικία με σχετικά μικρό αριθμό μυρμηγκιών δεν δύναται να εντοπίσει και να εκμεταλλευτεί μια πηγή τροφής καθώς η αρνητική ανάδραση εμποδίζει το σχηματισμό τέτοιων δομών.

Ορισμένες βασικές ιδιότητες που χαρακτηρίζουν αυτο-οργανωμένα συστήματα είναι[6]:

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

- Εν δυνάμει εμφάνιση πολλών σταθερών και αποδεκτών καταστάσεων (multistable). Η ενίσχυση των τυχαίων αποκλίσεων δημιουργεί τις δομές και συνεπώς αν μια από αυτές τις τυχαίες αποκλίσεις ενισχυθεί, τότε το συστήμα θα οδηγηθεί σε σύγκλιση με μία κατάσταση που θα έχει προκύψει βάση των αρχικών συνθηκών. Ως παράδειγμα αναφέρουμε τη ύπαρξη δύο ίσων προς την ποσότητα και την ποιότητα πηγών τροφής που έχουν την ίδια απόσταση από τη φωλιά. Ενώ και οι δυο πηγές έχουν την ίδια πιθανότητα εκμετάλλευσης, τελικά μόνο μία από αυτές επιλέγεται. Το ποια από τις δύο θα επιλεγεί εξαρτάται από

[6] Bonabeau, E., Dorigo, M., Theraulaz, G., (1999). Swann Intelligence, From Natural to Artificial Systems, Oxford University Press.

τις αρχικές τυχαίες διακυμάνσεις. Δηλαδή το σύστημα τελικά συγκλίνει στην ολική εκμετάλλευση μίας εκ των δύο, ενώ και οι δύο λύσεις είναι αποδεκτές.

- Η δημιουργία διακλαδώσεων (bifurcation) λόγω της μεταβολής μερικών παραμέτρων. Ως διακλάδωση εννοούμε το πέρασμα από μια φάση δραστηριότητας σε μια άλλη. Για παράδειγμα το είδος τερμίτη[7] Bellicositermes Natalensis et Cubitermes κατασκευάζει τις φωλιές του σε δύο διακριτές διαδοχικές φάσεις. Οι τερμίτες χρησιμοποιούν χωμάτινα σφαιρίδια εμποτισμένα με φερομόνη τα οποία εναποθέτουν στο έδαφος. Κατά την αρχική αυτή φάση η εναπόθεση γίνεται τυχαία και μη συντονισμένα (non coordinated). Αυτή η φάση διαρκεί μέχρι, για τυχαίους λόγους, μια από τις εναποθέσεις σφαιριδίων χώματος φτάσει σε ένα κρίσιμο μέγεθος. Δηλαδή να υπάρξει μια συγκεκριμένη συγκέντρωση φερομόνης σε ένα σημείο. Τότε το σύστημα περνά στη δεύτερη φάση, τη φάση συντονισμού (coordination phase). Η ομάδα των 'οικοδόμων' αυξάνεται και αρχίζουν να δημιουργούνται στήλες ή λωρίδες. Με το πέρασμα του χρόνου, μέσω της θετικής ανάδρασης, θα δημιουργηθούν και τόξα που ενώνουν τις στήλες μεταξύ τους και η φωλιά θα έχει ολοκληρωθεί. Αν ο αριθμός των τερμιτών είναι μικρός, η φερομόνη εξατμίζεται και τότε ποτέ κανένας σχηματισμός σφαιριδίων δεν ξεπερνά το όριο της φερομόνης που απαιτείται για να λειτουργήσει ο μηχανισμός της ενίσχυσης. Σε αυτή την περίπτωση παρατηρούμε μονό τη μη συντονισμένη φάση.

[7] Grassé, P. P., (1959). La Reconstruction du nid et les coordinations interindividuelles chez, *Bellicositermes Natalensis et Cubitermes sp.* La théorie de la stigmergie: Essai d' interprétation du comportement des termites constructeurs. *Insectes-Sociaux.*, 6, 41-80.

3.4 Στιγμεργία

Η ετυμολογία του όρου είναι ελληνική καθώς συντίθεται από τις λέξεις έργο (ergo) και στίγμα (stigma) και προτάθηκε το 1959 από τον Grassé[8]. Ο όρος θέλει να εκφράσει την ιδέα ότι οι πράξεις ενός ατόμου αφήνουν στίγματα στο περιβάλλον και ότι αυτά τα στίγματα γίνονται αντιληπτά από το άτομο και τους ομοίους του και προσαρμόζουν ανάλογα τις μελλοντικές τους δράσεις. Είναι μέθοδος έμμεσης επικοινωνίας σε αποικίες κοινωνικών εντομών που στηρίζεται στην τροποποίηση του περιβάλλοντος για τη μεταφορά του μηνύματος.

Ο ακριβής ορισμός του Grassé για τη στιγμεργία είναι:

'Διέγερση των εργατών-μυρμηγκιών βάση της απόδοσης που επέτυχαν.'

Ο Grasse έδειξε ότι ο συντονισμός του χτισίματος της φωλιάς δεν εξαρτάται από καθένα εργάτη ξεχωριστά, αλλά είναι διαδικασία που επιτυγχάνεται από την αποικία ως σύνολο. Μια διάταξη του περιβάλλοντος προκαλεί ένα ερέθισμα που κάνει τον εργάτη-τέρμιτη να αντιδράσει, μετασχηματίζοντας τη διάταξη του περιβάλλοντος σε μια νέα που προκαλεί νέα ερεθίσματα. Στα νέα αυτά ερεθίσματα θα αντιδράσουν, ίσως με νέο διαφορετικό από πριν τρόπο, τα μέλη της αποικίας συμπεριλαμβανομένου και του ίδιου εργάτη-τερμίτη

[8] Grassé, P. P., (1959). La Reconstruction du nid et les coordinations interindividuelles chez, *Bellicositermes Natalensis et Cubitermes sp.* La théorie de la stigmergie: Essai d' interprétation du comportement des termites constructeurs. *Insectes-Sociaux.*, 6, 41-80.

Ο Bonabeau[9] υποστηρίζει ότι, το πρόβλημα με τον ορισμό του Grasse είναι ότι δεν απαντά στο σημαντικό ερώτημα του πώς πρέπει να οργανωθούν τα ερεθίσματα στο χώρο και στο χρόνο αναζήτησης. Αυτή η ελαφρά ασάφεια δεν ελαττώνει τη σημαντικότητα της ιδέας της στιγμεργίας που μας δίνει μια εκ των έσω εικόνα για τη εξέλιξη και διατήρηση πολύπλοκων αναδυόμενων συστημάτων.

Η στιγμεργία είναι ένας αποκεντρωμένος μηχανισμός ροής πληροφοριών στο οποίο τα άτομα, μέρη του συστήματος, επικοινωνούν μέσω των αλλαγών που προκαλούν στο τοπικό τους περιβάλλον

Οι Dorigo,Bonabeau και Theraulaz[10] επαύξησαν το παλιό ορισμό ως εξής:

'Στιγμεργετική επικοινωνία είναι κάθε έμμεση επικοινωνία η οποία πραγματοποιείται με φυσικές μεταβολές στο περιβάλλον οι οποίες είναι μόνο τοπικά προσπελάσιμες από τα επικοινωνούντα άτομα.'

Τα μυρμήγκια έχουν ένα αδένα που παράγει τη χημική ουσία φερομόνη και μέσω αυτής της ουσίας υλοποιείται η έμμεση επικοινωνία που οδηγεί στην αλληλεπίδραση των ατόμων της αποικίας.

Η έλλειψη φερομόνης στο χώρο αναζήτησης θα οδηγήσει τα μυρμήγκια να κάνουν μια τυχαία έρευνα του περιβάλλοντος τους. Η ανακάλυψη μιας πηγής τροφής θα συνδυαστεί με την εναπόθεση φερομόνης στη διαδρομή της επιστροφής προς τη

[9] Bonabeau, E., Dorigo, M., and Theraulaz, G. (2000). Inspiration for optimization from social insect behaviour. *Nature*, 406:39.42.
[10] Dorigo M., Bonabeau E., Theraulaz G., Ant algorithms and stigmergy, Future Generation Computer Systems 16 (2000) 851-871

φωλιά. Έτσι δημιουργείται ένα μονοπάτι-ίχνος φερομόνης. Αν ο αριθμός των μελών της αποικίας δεν είναι ικανός να συντηρήσει το μονοπάτι, με ένα σημαντικό αριθμό διαβάσεων που σημαίνει και εναπόθεση ποσοτήτων φερομόνης, τότε η εξάτμιση της φερομόνης θα εξαφανίσει το μονοπάτι.

Η επικοινωνία μέσω στιγμεργίας κληροδοτεί σε αυτά τα βιολογικά συστήματα εγγενή πλεονεκτήματα ευελιξίας και ευρωστίας. Κάθε μεταβολή του περιβάλλοντος είτε από εσωτερικούς, είτε από εξωτερικούς παράγοντες θεωρείται ως μεταβολή που προκλήθηκε από άλλα μέλη της αποικίας. Συνεπώς τα μέλη της αποικίας αντιδρούν όπως και πριν, απλώς τώρα λαμβάνουν υπόψη τα νέα δεδομένα. Αυτή η συμπεριφορά είναι ιδιαίτερα χρήσιμη στο τεχνητό αντίστοιχο αυτού του βιολογικού συστήματος καθώς τα τεχνητά μέλη της αποικίας δεν θα χρειάζονται επαναπρογραμματισμό και ο υπάρχον αλγόριθμος θα επαρκεί για την αντιμετώπιση των νέων δεδομένων. Η ιδέα πίσω από τους αλγορίθμους μυρμηγκιών είναι να καταφέρουμε να δημιουργήσουμε τεχνητή στιγμεργία ώστε να συντονίσουμε κοινωνίες τεχνητών μυρμηγκιών.

3.5 Η Βιολογική Προέλευση των Ιδεών

Οι Goss[11] και Deneubourg[12] δημοσιεύουν το 1989 και το 1990 τις πρώτες εργασίες σχετικά με τον τρόπο αναζήτησης τροφής των μυρμηγκιών μέσα στο χώρο. Σε αυτές τις εργασίες αναφέρεται για πρώτη φορά ότι μεταξύ δύο μονοπατιών (trails) που οδηγούν στην πηγή της τροφής, σε βάθος χρόνου, ο πληθυσμός της αποικίας θα συγκλίνει προς το συντομότερο μονοπάτι. Επισημαίνεται ακόμα, ότι η πιθανότητα να

[11] Goss, S., Aron S., Deneubourg, J.L., Pasteels, J.M., (1989). Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76, p. 579-581.

[12] Deneubourg, J.-L., Aron, S. Goss, S. Pasteels, J.M., (1990). The Self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behaviour, 3, 159-168.

συγκλίνουν προς το συντομότερο μονοπάτι αυξάνεται καθώς αυξάνεται η διαφορά σε μήκος μεταξύ των δύο επιλογών. Ως εξήγηση για αυτή τη συμπεριφορά προτείνεται η θετική ανάδραση (Positive Feedback) και το διαφοροποιημένο μήκος μιας διαδρομής (Differential Path Length). Μαζί αυτές οι δύο ιδέες σχηματίζουν έναν τρόπο έμμεσης επικοινωνίας (indirect means of communication) που πρακτικά είναι *στιγμεργία* (Stigmergy).

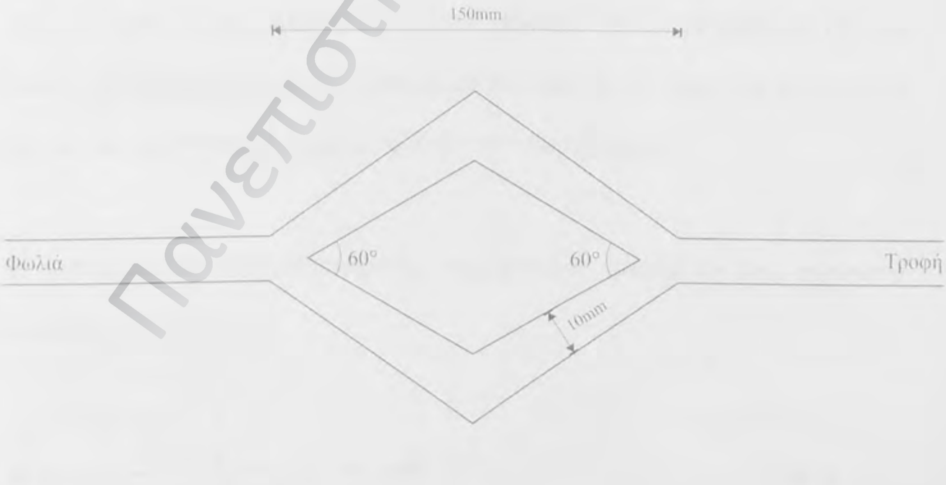
Τα Πειράματα με Διπλή Γέφυρα

Αρχικά χρησιμοποιήθηκε ένα είδος μυρμηγκιών με σχετικά περιορισμένες ικανότητες προσανατολισμού και φερομόνη που έχει σχετικά μεγάλο χρόνο ημιζωής (περίπου 30 λεπτά) από την Αργεντινή (Iridomyrmex humilis) τα οποία τοποθετούνταν σε μια πειραματική διάταξη όπου στην αρχή μιας γέφυρας έπρεπε να επιλέξουν ανάμεσα σε δύο διαδρομές ίσου μήκους για μια συγκεκριμένη χρονική περίοδο.

Στη πειραματική διάταξη ορίζεται ως l_l το μήκος του μακρύτερου κλάδου της γέφυρας και l_s το μήκος του μικρότερου κλάδου. Ο λόγος $r = \frac{l_l}{l_s}$ αποτελεί το ρυθμιστικό παράγοντα.

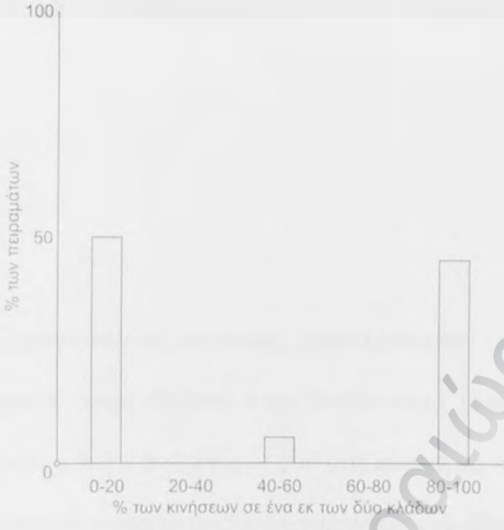
Παρατηρήθηκε ότι η πιθανότητα ένα μυρμήγκι να επιλέξει τον κλάδο που επέλεγαν τα περισσότερα μυρμήγκια αυξανόταν ταχύτατα και μη γραμμικά και αυτό κυρίως λόγω των άλλων που είχαν προηγηθεί. Μια άλλη σημαντική παρατήρηση ήταν ότι ένα μυρμήγκι από την αναχώρηση μέχρι και την επιστροφή του στη φωλιά εναποθέτει ένα μέσο ποσό φερομόνης στη διαδρομή που ακολουθεί. Συνεπώς η επιλογή μιας διαδρομής είναι θέμα και του αριθμού των μυρμηγκιών που υπάρχουν στο πείραμα και έχουν περάσει από τη συγκεκριμένη διαδρομή και όχι μόνο του συντομότερου χρόνου.

Στο πρώτο πείραμα με $r = 1$, δηλαδή με ίσο μήκος για τους δύο κλάδους αφήνονται ελεύθερα τα μυρμήγκια να κινηθούν μεταξύ φωλιάς και τροφής. Το πρώτο διάστημα δεν παρατηρείται κάποιο πρότυπο και τα μυρμήγκια κινούνται τυχαία. Τελικά όμως επιλέγουν έναν από τους δύο κλάδους μαζικά. Το γεγονός αυτό οφείλεται στο ότι όταν το πείραμα ξεκινά και οι δύο κλάδοι έχουν μηδενικές εναποθέσεις φερομόνης. Συνεπώς τα πρώτα μυρμήγκια θα επιλέξουν τυχαία ανάμεσα στους δύο κλάδους. Λόγω της ενίσχυσης τυχαίων διακυμάνσεων τελικά ένας εκ των δύο κλάδων θα αποκτήσει μεγαλύτερο φορτίο φερομόνης. Όταν συμβεί αυτό η θετική ανάδραση θα άγει την αποικία στον συγκεκριμένο κλάδο.



Σχήμα 3.1

Η πειραματική διάταξή με κλάδους ίσου μήκους. Προσαρμογή από Goss (1989).



Σχήμα 3.2 Ποσοστό κινήσεων στη πειραματική διάταξή με κλάδους ίσου μήκους. Προσαρμογή από Goss (1989).

Στο σχήμα 3.1 και στο γράφημα στο σχήμα 3.3 απεικονίζεται η πειραματική διάταξη διπλής γέφυρας με ίσο μήκος για τους δύο κλάδους (r=1) που εφαρμόζεται από τον Goss[13]. Το γράφημα δείχνει τη συμπεριφορά των μυρμηγκιών όπου επιλέγουν τελικά είτε τον ένα είτε τον άλλο κλάδο με περίπου την ίδια πιθανότητα.

Η μοντελοποίηση της εξερευνητικής συμπεριφοράς εκφράζεται στις παρακάτω εξισώσεις.

$$p(A) = \frac{(20 + A_i)^2}{(20 + A_i)^2 + (20 + B_i)^2} = 1 - p(B) \qquad (3.1)$$

[13] Goss, S., Aron S., Deneubourg, J.L., Pasteels, J.M., (1989). Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76, p. 579-581.

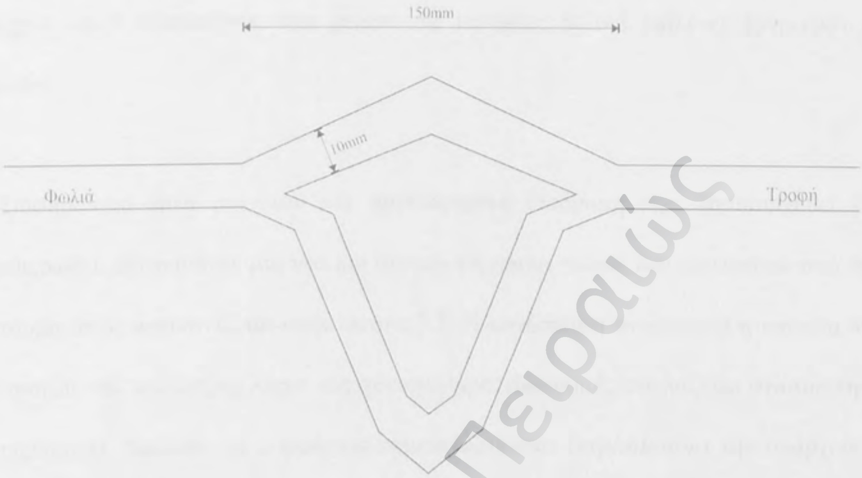
$$A_{i+1} = A_i + \delta \qquad (3.2)$$

$$B_{i+1} = B_i + (1-\delta) \qquad (3.3)$$

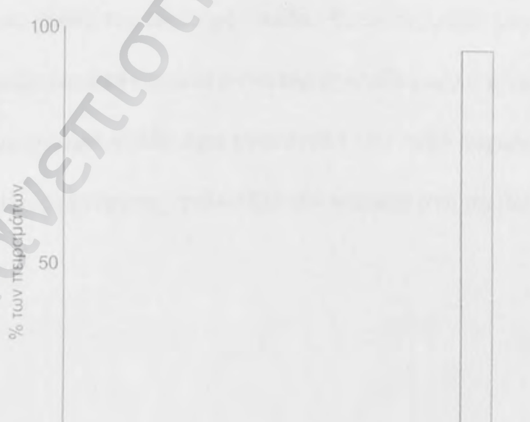
Αφού i μυρμήγκια έχουν διασχίσει τη γέφυρα, έχουν εναποτεθεί i μονάδες φερομόνης εκ των οποίων A_i και B_i στους κλάδους A και B αντίστοιχα. Η εξίσωση 3.1 δίνει τη πιθανότητα να επιλεγεί ο A ή ο B κλάδος με βάση τις ποσότητες A_i και B_i. Μετά την επιλογή, στον επιλεχθέντα κλάδο εναποτίθεται η φερομόνη με βάση τις εξισώσεις 3.2 και 3.3. Πρέπει να αναφέρουμε ότι το δ είναι στοχαστική μεταβλητή και λαμβάνει τιμές 0 ή 1 με πιθανότητα p(A) ή p(B). Επίσης αναφέρουμε ότι οι τιμές 2 για τον εκθέτη και 20 για το συντελεστή άρμοζαν με τα αποτελέσματα του συγκεκριμένου πειράματος.

Κατά τη δεύτερη εκτέλεση του πειράματος, δύο άνισες διαδρομές δημιουργήθηκαν όπως απεικονίζεται στο σχήμα 3.3 . Ο ένας εκ των δύο κλάδων είχε διπλάσιο μήκος, δηλαδή $r = 2$. Τα αποτελέσματα έδειξαν ότι γινόταν όλο και πιθανότερο να επιλεγεί ο πιο σύντομος κλάδος καθώς αυξανόταν η διαφορά ανάμεσα στους δυο κλάδους.

Κατά το πείραμα με $r = 2$ τα μυρμήγκια κινούνται μεταξύ φωλιάς και τροφής. Όπως αναμένεται στην πρώτη διακλάδωση επιλέγουν τυχαία και περίπου με ίση αναλογία. Η διαφορά με το προηγούμενο πείραμα έγκειται στο ότι τα μυρμήγκια που επέλεξαν το κοντό κλάδο φτάνουν πρώτα στην τροφή και ξεκινούν τη διαδρομή της επιστροφής προς τη φωλιά. Στη διαδρομή αυτή θα πρέπει πάλι να επιλέξουν μεταξύ μικρού και μεγάλου κλάδου. Το μεγαλύτερο ίχνος φερομόνης στον μικρό κλάδο θα οδηγήσει στο να τον προτιμήσουν. Συνεπώς όλο και μεγαλύτερο ίχνος θα συσσωρεύεται στο κοντό κλάδο και το σύστημα θα συγκλίνει προς αυτόν.



Σχήμα 3.3 Η πειραματική διάταξή όπου ο ένας κλάδος έχει διπλάσιο μήκος από τον άλλο. Προσαρμογή από Goss (1989).

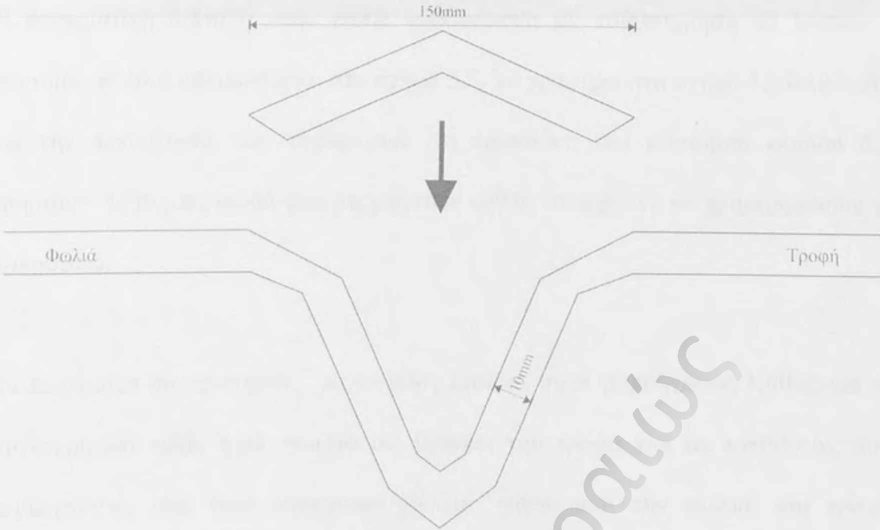


Σχήμα 3.4 Ποσοστό κινήσεων στη πειραματική διάταξή όπου ο ένας κλάδος έχει διπλάσιο μήκος από τον άλλο. Προσαρμογή από Goss (1989).

Η πειραματική διάταξη διπλής γέφυρας με διπλάσιο μήκος για ένα εκ των δύο κλάδων (r=2) απεικονίζεται στο σχήμα 2.2. Το γράφημα σε αυτή την περίπτωση δείχνει ότι η πλειοψηφία των μελών της αποικίας τελικά επιλέγει το μικρότερο κλάδο.

Αξιοσημείωτο είναι ότι όταν μια συγκεκριμένη διαδρομή είχε δημιουργηθεί και καθιερωθεί, προτεινόταν μια νέα πιο σύντομη η οποία τελικά δεν επιλεγόταν από την αποικία όπως απεικονίζεται στην εικόνα 2.3. Η κατάσταση στην οποία η αποικία δεν εντοπίζει νέες καλύτερες λύσει ως συντομότερες διαδρομές ονομάζεται στασιμότητα (stagnation). Δηλαδή τα μυρμήγκια εκμεταλλεύονται (*exploitation*) την υπάρχουσα λύση και δεν εξερευνούν (*exploration*) την νέα καλύτερη λύση που τους είχε προταθεί.

Το πείραμα ξεκινά χωρίς τον σύντομο κλάδο. Τριάντα λεπτά μετά την έναρξη του πειράματος προσφέρεται στη αποικία ο σύντομος κλάδος αλλά η κατάσταση έχει είδη παγιωθεί. Στον μακρύτερο κλάδο έχει εναποτεθεί ένα πολύ ισχυρό ίχνος φερομόνης και ο χαμηλός ρυθμός εξάτμισης εγκλωβίζει την αποικία στη μη βέλτιστη επιλογή.



Σχήμα 3.5 Η πειραματική διάταξή όπου αρχικά προσφέρεται ο μακρύς κλάδος και ύστερα ο σύντομος. Προσαρμογή από Goss (1989).

Σχήμα 3.6 Ποσοστό κινήσεων στη πειραματική διάταξή όπου αρχικά προσφέρεται ο μακρύς κλάδος και ύστερα ο σύντομος. Προσαρμογή από Goss (1989).

Η πειραματική διάταξη στην οποία προσφέρεται με καθυστέρηση 30 λεπτών ο σύντομος κλάδος απεικονίζεται στο σχήμα 3.5. Το γράφημα στο σχήμα 3.6 δείχνει ότι για την πλειοψηφία των πειραμάτων η προσθήκη του σύντομου κλάδου δεν επηρεάζει τη συμπεριφορά των μυρμηγκιών καθώς συνεχίζουν να χρησιμοποιούν το μακρύτερο.

Τα πειράματα συνεχίστηκαν[14] με το είδος Lasious niger μυρμηγκιών. Καθώς για το συγκεκριμένο είδος ήταν γνωστό ότι βρίσκει την τροφή του σε τοποθεσίες που περικλείονται από έναν θεωρητικό κύκλο γύρω από την φωλιά, και τροφή τοποθετήθηκε σε διάφορες τοποθεσίες γύρω από την φωλιά. Παρατηρήθηκε ότι καθώς η ποσότητα τροφής σε μία τοποθεσία μειωνόταν, όμοια μειωνόταν και ο αριθμός των μυρμηγκιών που κατευθύνονταν στην περιοχή. Προφανώς η εξάτμιση της φερομόνης συνέβαλε στη διαδικασία της αρνητικής ανάδρασης (negative feedback) και στην τελική αχρηστία της διαδρομής που οδηγεί στη συγκεκριμένη πηγή τροφής. Ο σκοπός του πειράματος να εξετάσει αν ισχύουν οι αρχές που παρατηρήθηκαν στα προηγούμενα πειράματα επιτεύχθηκε καθώς ανάλογα με τον αριθμό των μυρμηγκιών και των πηγών τροφής οι ενεργές διαδρομές, που ορίζονται από την φερομόνη που υπάρχει σε αυτές, άλλαζαν τριγύρω από την αποικία.

Οι εξισώσεις που προέκυψαν από αυτό το μοντέλο είναι οι παρακάτω:

$$\frac{dS_i}{dt} = \Phi - g \cdot \frac{F_i S_i}{(a + S_i)} - r \cdot S_i \qquad (3.4)$$

---

[14] Beckers, R., Deneubourg, J.L., Goss, S. , (1992). Trails and U-turns in the Selection of a Path by the Ant Lasius niger, Journal of Theoretical Biology, vol 159, p.397-415.

με $i = 1,...,b$

$$\frac{dC_i}{dt} = g \cdot \frac{F_i S_i}{(a + S_i)} - e \cdot C_i \tag{3.5}$$

$$F_i = N((1-2q) \cdot f_i + q \cdot f_{i+1} + q \cdot f_{i-1}) \tag{3.6}$$

$$F_i = \frac{(20 + C_i)^2}{\sum (20 + C_j)^2} \tag{3.7}$$

με $\sum f_i = 1$

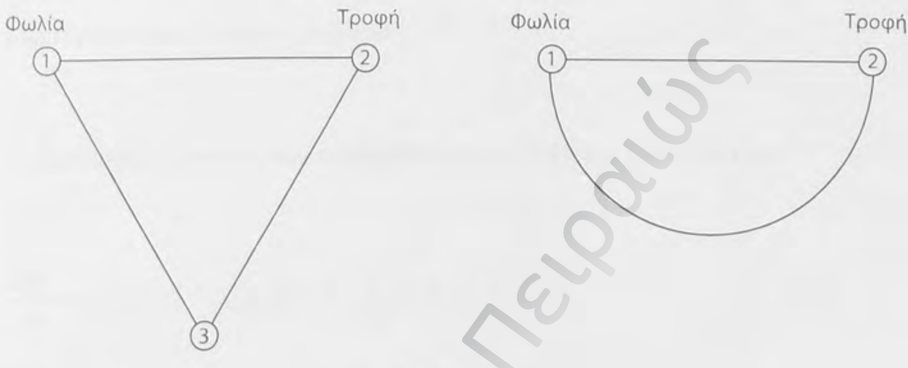
Η περιοχή αναζήτησης τροφής γύρω από την αποικία χωρίστηκε σε τομείς (sectors) b και ορίζεται ως Φ_i το ποσό τροφής που φτάνει ανά μονάδα χρόνου σε κάθε τομέα, ο καθένας εκ των οποίων περιέχει S_i ποσό τροφής. Επίσης ένα ποσοστό r της τροφής χάνεται ανά μονάδα χρόνου. Υποθέτουμε ότι σε κάθε τομέα οδηγεί μια διαδρομή με C μονάδες φερομόνης πάνω σε αυτή και εξ αυτής ένα ποσοστό e, εξατμίζεται ανά μονάδα χρόνου.

Τα μυρμήγκια που φεύγουν από τη φωλιά ανά μονάδα χρόνου για αναζήτηση τροφής ορίζονται ως N. Πιθανοκρατικά ένα ποσοστό f_i επιλέγει τον τομέα i με βάση τη γενική μορφή της εξίσωσης 2.1 και ένα άλλο μικρότερο ποσοστό q διαμοιράζεται στους γειτονικούς τομείς.

Το ποσό τροφής που υπάρχει μια δεδομένη χρονική στιγμή σε ένα τομέα δίδεται από το κλάσμα $g \cdot \frac{F_i S_i}{(a+S_i)}$. Όπου g και a είναι πειραματικές σταθερές, Si το ποσό τροφής και F_i ο αριθμός των μυρμηγκιών που αναζητούν τροφή στον τομέα i την συγκεκριμένη χρονική στιγμή. Στο τέλος κάθε επανάληψης όλα τα μυρμήγκια επιστρέφουν στη φωλιά *αλλά* μόνο αυτά που βρήκαν τροφή επιτρέπεται να εναποθέσουν φερομόνη μια μονάδα φερομόνης στη διαδρομή που ακολούθησαν.

Μετάβαση από το Βιολογικό στο Τεχνητό Μοντέλο

Αυτές οι πρώτες προσπάθειες να μοντελοποιηθεί αυτή η βιολογική διαδικασία ήταν η αρχή των Αλγορίθμων Μυρμηγκιών. Το μοντέλο προσπαθεί να προσομοιώσει το πρόβλημα και τις διαδικασίες με υψηλή πιστότητα. Το πρόβλημα ορίζεται ως ένα γράφημα μέσα στο οποίο κυκλοφορούν τα μυρμήγκια. Κάθε μυρμήγκι έχει σκοπό να περιδιαβεί πιθανοκρατικά (probabilistic) το γράφημα με σκοπό να κάνει μια διαδρομή (trail). Τα μυρμήγκια μπορούν να σταλούν στο γράφημα είτε σε σειρά (ένα τη φορά) είτε παράλληλα (όλα μαζί) και στο τέλος της διαδρομής ή ενός βήματος (step) του αλγορίθμου θα απεικονίσουν τη λύση που βρήκαν στο Πίνακα Φερομόνης (Pheromone Matrix). Κάθε στοιχείο αυτού του πίνακα δείχνει τη σύνδεση που υπάρχει ανάμεσα σε δύο τμήματα του γραφήματος και πόσο δυνατή είναι αυτή η σύνδεση και περιγράφεται με τον όρο 'ένταση διαδρομής' (trail intensity). Για να αποκτήσουν δικαίωμα σε αυτή την απεικόνιση του Πίνακα Φερομόνης πρέπει να συμμορφώνονται με τον Κανόνα Ανανέωσης Διαδρομής (Trail Update Rule), όπως αυτός ορίζεται στην εκάστοτε υλοποίηση του αλγορίθμου. Δηλαδή η λειτουργία αυτού του πίνακα είναι να αποθηκεύει τις λύσεις που συνήθως σχετίζονται άμεσα με τις τιμές της αντικειμενικής συνάρτησης, αλλά και να ελαχιστοποιεί στοιχεία που δεν περιέχονται στις λύσεις που προκρίθηκαν από τον κανόνα ανανέωσης. Αυτή η ελαχιστοποίηση ονομάζεται παρακμή λύσεων (decay) και είναι μέρος της διαδικασίας να ευρεθεί η βέλτιστη λύση.



Σχήμα 3.7

Η πειραματική διάταξη διπλής γέφυρας με άνισους κλάδους ισοδυναμεί με τα παραπάνω γραφήματα καθώς προσφέρεται σύντομη αλλά και μακριά διαδρομή από την φωλιά στην πηγή τροφής. Στο σχήμα στα αριστερά οι αποστάσεις ανάμεσα στους κόμβους είναι ίδιες, συνεπώς η μακρύτερη διαδρομή είναι διπλάσια άρα r=2. Στο σχήμα στα δεξιά θέτουμε ότι η μακρύτερη διαδρομή είναι r. Και στις δύο περιπτώσεις ένα μυρμήγκι που κινείται στο μακρύτερο κλάδο θα ανανεώσει τη φερομόνη r χρονικές μονάδες αργότερα.

Η γενική μορφή της εξίσωσης (3.1) που προκύπτει από τα πειράματα του Deneubourg δίδεται παρακάτω:

$$p_{is}(t) = \frac{(t_s + \phi_{is}(t))^a}{(t_s + \phi_{is}(t))^a + (t_s + \phi_{il}(t))^a} \qquad (3.8)$$

όπου με βάση τα πειράματα $a = 2$ και p_{il}, ώστε $p_{is}(t) + p_{il}(t) = 1$.

Το μοντέλο αυτό δεν λαμβάνει υπόψη του την εξάτμιση της φερομόνης και θεωρεί ότι η εναπόθεση φερομόνης στο κάθε κλάδο είναι ανάλογη του αριθμού των μυρμηγκιών που τον χρησιμοποίησαν.

Οι διαφορικές εξισώσεις που περιγράφουν την εξέλιξη του μοντέλου είναι:

$$\frac{d\phi_{is}}{dt} = \psi\, p_{js}(t - t_s) + \psi\, p_{is}(t), \quad (i = 1, j = 2,; i = 2, j = 1) \tag{3.9}$$

$$\frac{d\phi_{il}}{dt} = \psi\, p_{jl}(t - t_s) + \psi\, p_{il}(t), \quad (i = 1, j = 2,; i = 2, j = 1) \tag{3.10}$$

Το παραπάνω μοντέλο έχει προσομοιωθεί από τον Liu[15] με τη χρήση της μεθόδου Monte Carlo. Για κάθε πείραμα με ίσο ή άνισο κλάδο, δηλαδή με r=1 και r=2, έγιναν 1000 προσομοιωμένα πειράματα με ψ=0,5 μυρμήγκια ανά δευτερόλεπτο. Τα αποτελέσματα ήταν συνεπή με τα βιολογικά πειράματα. Δείχθηκε ότι και στα προσομοιωμένα πειράματα τα τεχνητά μυρμήγκια τελικά επέλεγαν ένα και των δύο κλάδων για πείραμα ίσων κλάδων και τη συντομότερη διαδρομή για πείραμα άνισων κλάδων. Σημαντικό είναι ότι σε αυτή την υλοποίηση τα μυρμήγκια εναποθέτουν φερομόνη στις διαδρομές προς, αλλά και από, την πηγή τροφής. Όπως και στο βιολογικό μοντέλο, αν δεν υπάρχει και στις δύο διαδρομές εναπόθεση φερομόνης, η αποικία δε συγκλίνει στην άριστη διαδρομή.

[15] Liu J.S. (2001). Monte Carlo Strategies on Scientific Computing. New York, Springer-Verlag

Με μεγαλύτερη σαφήνεια η μετάβαση από το βιολογικό στο τεχνητό μοντέλο παρουσιάζεται παρακάτω[16].

Έστω ένα γράφημα με δύο κόμβους όπου η σύνδεση ανάμεσα τους γίνεται με ένα μακρύ και ένα κοντό τόξο. Η σχέση μεταξύ των τόξων ορίζεται από το λόγο r και ο χρόνος στο σύστημα είναι διακριτός. Σε κάθε χρονική μονάδα ένα μυρμήγκι κινείται προς το γειτονικό του κόμβο με σταθερή ταχύτητα ίση με μία μονάδα μήκους ανά χρονική μονάδα. Η κίνηση αυτή εναποθέτει μία μονάδα φερομόνης στον χρησιμοποιημένο τόξο. Τα μυρμήγκια κινούνται πιθανοκρατικά.

Τότε ορίζουμε τις πιθανότητες $p_{is}(t)$ και $p_{il}(t)$ να κινηθεί ένα μυρμήγκι από τον κόμβο i, $\{i \in 1,2\}$, ως:

$$p_{is}(t) = \frac{[\phi_{is}(t)]^{a}}{[\phi_{is}(t)]^{a} + [\phi_{il}(t)]^{a}}, \quad p_{il}(t) = \frac{[\phi_{il}(t)]^{a}}{[\phi_{is}(t)]^{a} + [\phi_{il}(t)]^{a}} \tag{3.11}$$

Η ανανέωση της φερομόνης γίνεται βάση των παρακάτω εξισώσεων.

$$\phi_{is}(t) = \phi_{is}(t-1) + p_{is}(t-1)m_{i}(t-1) + p_{js}(t-1)m_{j}(t-1) \tag{3.12}$$

Με $(i=1, j=2,; i=2, j=1)$

$$\phi_{il}(t) = \phi_{il}(t-1) + p_{il}(t-1)m_{i}(t-1) + p_{jl}(t-r)m_{j}(t-r) \tag{3.13}$$

Με $(i=1, j=2,; i=2, j=1)$

[16] Dorigo, M., Stützle, T., (2004). Ant Colony Optimization, MIT Press, Cambridge, Massachusetts

Όπου:

$p_{is}(t)$ η πιθανότητα ένα μυρμήγκι στο κόμβο i την χρονική στιγμή t να επιλέξει το μικρότερο κλάδο

$p_{il}(t)$ η πιθανότητα ένα μυρμήγκι στο κόμβο i την χρονική στιγμή t να επιλέξει το μεγαλύτερο κλάδο

$\phi_{is}(t), \phi_{il}(t)$ το ίχνος φερομόνης στο κοντό και μακρύ κλάδο αντίστοιχα

$m_i(t)$ ο αριθμός των μυρμηγκιών στο κόμβο i την χρονική στιγμή t και δίδεται από τον τύπο

$$m_i(t) = p_{js}(t-1)m_j(t-1) + p_{jl}(t-r)m_j(t-r), \quad (i=1, j=2,; i=2, j=1)$$

r ένας ακέραιος που περιγράφει το λόγο του μήκους των τόξων και δίδεται από τον τύπο $r = \frac{l_l}{l_s}$

Το παραπάνω μοντέλο δεν λαμβάνει υπόψη τη στοχαστική συμπεριφορά του ενός μυρμηγκιού αλλά τη μέση συμπεριφορά του συστήματος. Επίσης λειτουργεί σε διακριτό χρόνο, κάτι που φαίνεται από το γεγονός ότι χρησιμοποιεί εξισώσεις διαφορών αντί για διαφορικές εξισώσεις.

Σύμφωνα με τους Dorigo και Di Caro[17] οι βασικές ομοιότητες και διαφορές στις έννοιες και ιδέες μεταξύ βιολογικού και τεχνητού μοντέλου μπορούν να συνοψιστούν στα παρακάτω:

[17] Dorigo M., DiCaro G. Gambardella J.L., (1999), Ant algorithm for discrete optimization, Artificial Life 5 (2), p 137-172

Αποικία Συνεργαζόμενων Ατόμων

Τα πραγματικά μυρμήγκια ζουν σε αποικίες όπως και τα τεχνητά που είναι μέρος ενός πληθυσμού ή μιας αποικίας από αυτόνομες σύγχρονες ή ασύγχρονες οντότητες. Οι οντότητες αυτές συνεργάζονται σε υψηλό επίπεδο (global scale) για να επιλύσουν το πρόβλημα δημιουργώντας μια δομή. Ως δομή εννοούμε τη βέλτιστη λύση που προκύπτει από την τοπική αλληλεπίδραση των τεχνητών μυρμηγκιών.

Ίχνος Φερομόνης και Στιγμεργία

Τα πραγματικά μυρμήγκια επικοινωνούν μεταξύ τους έμμεσα και αυτό γίνεται τροποποιώντας το περιβάλλον τους. Αυτή η ιδέα στιγμεργετικής επικοινωνίας υπάρχει και στα τεχνητά μυρμήγκια όπου χρησιμοποιούν ως μνήμη μιας διαδρομής μια αριθμητική τιμή στη θέση της φερομόνης. Η αριθμητική τιμή απεικονίζεται στον πίνακα φερομόνης όπου για κάθε κόμβο που χρησιμοποιήθηκε εναποτίθεται τεχνητή φερομόνη. Η τιμή αυτή όπως και στα βιολογικά μυρμήγκια εξαρτάται από την επίδοση και/ή την ιστορία του τεχνητού μυρμηγκιού και είναι διαθέσιμη σε όλα τα μέλη της αποικίας. Στην πραγματικότητα η φερομόνη εξατμίζεται αλλά και αυτό έχει ενσωματωθεί στο μοντέλο με μια πρόβλεψη μείωσης της τιμής της. Η εξάτμιση ενσωματώνει την ιδέα της αρνητικής ανάδρασης ώστε η αποικία σιγά σιγά να ξεχνά παλιές λύσεις και να μην εγκλωβίζεται σε τοπικά βέλτιστα.

Αναζήτηση Ελάχιστης Διαδρομής και Τοπικές Κινήσεις

Τα τεχνητά μυρμήγκια έχουν ως στόχο την εύρεση την ελάχιστης διαδρομής, δηλαδή το ελάχιστο κόστος, μεταξύ αφετηρίας (φωλιά) και τερματισμού (πηγή τροφής). Στα πειράματα τα πραγματικά μυρμήγκια κινούνται σε γειτνιάζουσες περιοχές και δεν μπορούν να βρεθούν ξαφνικά σε άλλες. Στα τεχνητά το τι ορίζεται ως γειτνιάζουσα περιοχή είναι σχετικό με το κάθε συγκεκριμένο πρόβλημα.

Στοχαστικές και Μυωπικές Πολιτικές Μετάβασης

Τα τεχνητά και τα βιολογικά μυρμήγκια στηρίζουν τις επιλογές τους στον πιθανοκρατίκο κανόνα μετάβασης με βάση αυστηρά τοπικές πληροφορίες για την κατάσταση του χώρου αναζήτησης. Οι τοπικές πληροφορίες είναι δύο ειδών. Το ένα είδος αφορά πληροφορίες που είναι γνωστές εκ των πρότερων (a priori). Στο βιολογικό μοντέλο για παράδειγμα τέτοιου είδους πληροφορία είναι η μορφολογία του εδάφους, κάτι που στο τεχνητό μοντέλο προσομοιώνεται με την ευριστική πληροφορία (heuristic information). Το άλλο είδος αφορά πληροφορίες που έχουν να κάνουν με τοπικές αλλαγές του περιβάλλοντος όπως τα μονοπάτια φερομόνης.

Τα παραπάνω τονίζουν τη συνάφεια μεταξύ βιολογικού και υπολογιστικού μοντέλου. Όμως πρέπει να τονίσουμε κάποιες ενστάσεις για το πώς περνάμε από ένα βιολογικό μοντέλο σε ένα υπολογιστικό.

Διακριτό Περιβάλλον εναντίον Συνεχούς Περιβάλλοντος

Το πραγματικό περιβάλλον στο βιολογικό μοντέλο είναι συνεχές και δυναμικό, ενώ τα προβλήματα στο υπολογιστικό μοντέλο συνήθως είναι διακριτά και στατικά.

Απλότητα του Προβλήματος

Τα προβλήματα που έλυναν τα μυρμήγκια στα πειράματα που περιγράψαμε είναι πολύ απλά. Οι μέθοδοι που χρησιμοποιούν τα μυρμήγκια μπορεί να μην είναι οι βέλτιστες ή σχετικές με ένα πολύπλοκο υπολογιστικό περιβάλλον.

Εφαρμογή της Στρατηγικής Επικοινωνίας

Το βιολογικό μοντέλο επικοινωνίας ίσως να μην είναι το βέλτιστο για να επικοινωνούν ανεξάρτητοι υπολογιστικοί πράκτορες μεταξύ τους, διότι αν και υπάρχει τρόπος αποθήκευσης της λύσης με αριθμητική αξία, η πληροφορία αυτή μπορεί και να μην είναι αριθμητική.

Κατανεμημένη Φύση

Στο βιολογικό μοντέλο υπάρχουν χιλιάδες μυρμήγκια που εναποθέτουν φερομόνη, ενώ σπάνια συμβαίνει αυτό στο υπολογιστικό καθώς ένα ή δυο μυρμήγκια ανα βήμα του αλγορίθμου επιτρέπονται να εναποθέσουν φερομόνη στον πίνακα που αποτελεί την μνήμη των λύσεων.

Σχετικότητα Περιβάλλοντος

Στη φύση το περιβάλλον βρίσκεται συνεχώς σε αλλαγή και η εξάτμιση της φερομόνης είναι συνεχής. Σε ένα υπολογιστικό περιβάλλον η παρακμή των λύσεων απεικονίζεται με την εξάτμιση και πιθανόν να υπάρχει κάποιος καλύτερος τρόπος να αντικαθιστούμε τις παλιές πληροφορίες.

3.6 Συμπεράσματα

Τα πρώτα πειράματα από το χώρο της βιολογίας για τον τρόπο αναζήτησης τροφής των μυρμηγκιών, μέσω των ιδεών της αυτο-οργάνωσης και της στιγμεργείας, έδειξαν τις πρώτες αρχές για το σχηματισμό πολύπλοκων λειτουργικών δομών σε συλλογικό επίπεδο αποικίας. Η Ευφυΐα Σμήνους είναι τεχνητή ευφυΐα που βασίζεται στη συλλογική συμπεριφορά αποκεντρωμένων, αυτο-οργανομένων συστημάτων. Ο πληθυσμός των ατόμων του συστήματος αποτελείται από άτομα που είτε είναι όμοια είτε διαφέρουν ελάχιστα μεταξύ τους αλλά είναι σχετικά απλά σαν μονάδες. Αυτό ο πληθυσμός αλληλεπιδρά δίχως να υπάρχει ένα γενικό πλαίσιο κανόνων ή συνειδητή συμπεριφορά των ατόμων και το σύνολο των αλληλεπιδράσεων εκφράζει τη συλλογική ευφυΐα του συστήματος. Αυτό είναι απόρροια της τοπικής αλληλεπίδρασης ατόμων που γειτνιάζουν στο χώρο και της συμπεριφοράς τους που ορίζεται από απλούς κανόνες. Συχνά αυτή η συμπεριφορά του κάθε ατόμου περιγράφεται με στοχαστικούς όρους που βασίζονται στη τοπική αντίληψη του ατόμου για το περιβάλλον του. Η Ευφυΐα Σμήνους περιλαμβάνει φυσικά και τεχνητά συστήματα που αποτελούνται από πολλά άτομα που συντονίζονται με αποκεντρωμένο έλεγχο και αυτο-οργάνωση. Συγκεκριμένα εστιάζει στις συλλογικές συμπεριφορές που αναδύονται από τις τοπικές αλληλεπιδράσεις των ατόμων μεταξύ τους και με το περιβάλλον. Παραδείγματα τέτοιων συστημάτων είναι οι αποικίες μυρμηγκιών και τερμιτών, σμήνη πουλιών, κοπάδια ψαριών, αγέλες ζώων και αποικίες μελισσών. Ανθρώπινες εφαρμογές με τέτοιες συμπεριφορές είναι πολύ-ρομποτικά συστήματα και λογισμικό με σκοπό τη βελτιστοποίηση.

Τα τέσσερα βασικά στοιχεία της αυτο-οργάνωσης, όπως η θετική ανάδραση, αρνητική ανάδραση, ενίσχυση τυχαίων διακυμάνσεων, αλλά και η ύπαρξη αλληλεπιδράσεων μεταξύ των ατόμων της αποικίας, δημιουργούν ένα σύστημα όπου το μεμονωμένο έντομο μπορεί να διεκπεραιώσει μόνο απλές λειτουργίες και δε λαμβάνεται υπόψη η περίπλοκη δομή του μεμονωμένου εντόμου. Η στιγμεργία είναι ένας αποκεντρωμένος μηχανισμός ροής πληροφοριών στο οποίο τα άτομα, μέρη του συστήματος, επικοινωνούν μέσω των αλλαγών που προκαλούν στο τοπικό τους περιβάλλον και στα κοινωνικά έντομα όπως τα μυρμήγκια υλοποιείται με την εναπόθεση στο έδαφος της χημικής ουσίας που ονομάζεται φερομόνη.

Οι παραπάνω ιδέες δημιουργούν ένα αρχικό πλαίσιο προσέγγισης για την κατανόηση αυτού δίχως κεντρική διοίκηση συστήματος των κοινωνικών εντόμων, που καταφέρνει να λειτουργεί αρμονικά σαν σύνολο ανακαλύπτοντας τη συντομότερη διαδρομή προς την πηγή τροφής.

Τα πειράματα με διπλή γέφυρα αναλύουν την εξερευνητική συμπεριφορά των μυρμηγκιών για την αναζήτηση τροφής και αποκαλύπτουν το ρόλο της φερομόνης στην επιλογή της διαδρομής προς την τροφή. Η μοντελοποίηση της εξερευνητικής συμπεριφοράς εκφράζεται στις πρωτόλειες εξισώσεις που δίδονται στις αρχές της δεκαετίας του 90 και αποτελούν την αφετηρία για τις περεταίρω μελέτες που τελικά έφεραν το ψηφιακό ανάλογο του μοντέλου αναζήτησης τροφής στις αποικίες μυρμηγκιών.

Η συνάφεια μεταξύ βιολογικού και υπολογιστικού μοντέλου ενισχύεται από το γεγονός ότι και στα δύο υπάρχει μια αποικία συνεργαζόμενων ατόμων, καθώς και ψηφιακή απεικόνιση του ίχνους φερομόνης, και μέσω αυτού της στιγμεργίας. Επίσης η αναζήτηση ελάχιστης διαδρομής, οι τοπικές κινήσεις αλλά και οι στοχαστικές και μυωπικές πολιτικές μετάβασης αποτελούν στοιχεία ομοιότητας μεταξύ των δύο μοντέλων.

Ως αντίλογος για το ψηφιακό υπολογιστικό μοντέλο αντιτάσσεται το διακριτό περιβάλλον του ψηφιακού μοντέλου εναντίον του συνεχούς περιβάλλοντος του βιολογικού αλλά και η απλότητα του προβλήματος που συναντάται στη φύση. Η σχετικότητα του περιβάλλοντος ως επιχείρημα, έχει σκοπό να αναδείξει ότι πιθανώς οι αναλογίες που χρησιμοποιούνται για την απεικόνιση των βιολογικών φαινομένων δεν είναι οι καλύτερες δυνατές. Το κύριο επιχείρημα για τη διαφορά μεταξύ των δύο μοντέλων έχει να κάνει με την εφαρμογή της στρατηγικής επικοινωνίας και την κατανεμημένη φύση του προβλήματος καθώς οι τομείς αυτοί αντιμετωπίζονται με σχετικά απλή προσέγγιση.

Με την ωρίμανση της μαθηματικής απεικόνισης αυτών των ιδεών γεννήθηκαν οι αρχικοί αλγόριθμοι μυρμηγκιών και παρουσιάστηκαν οι πρώτοι απτοί αλγόριθμοι που αντλούσαν την έμπνευση για τον τρόπο λειτουργίας τους από τον τρόπο λειτουργίας των κοινωνικών εντόμων που παρατηρήθηκε στις αποικίες μυρμηγκιών.

Κεφάλαιο 4

Η οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών

Εισαγωγή 4.1

Μια οικογένεια μεταευριστικών αλγορίθμων είναι ένα γενικό αλγοριθμικό πλαίσιο το οποίο δύναται να χρησιμοποιηθεί σε διαφορετικά προβλήματα βελτιστοποίησης με σχετικά λίγες τροποποιήσεις με σκοπό την προσαρμογή σε ένα συγκεκριμένο πρόβλημα. Σε αυτό το κεφάλαιο παρουσιάζεται η οικογένεια μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών (Ant Colony Optimization Algorithms), της οποίας η βασικές ιδέες εμπνέονται από τις ιδέες του κεφαλαίου 3. και πρόκειται για σύγχρονους μεταευριστικούς αλγορίθμους με ευρύτατο και πολύ επιτυχημένο πεδίο εφαρμογών. Στη πορεία του χρόνου οι αλγόριθμοι εμπλουτίστηκαν με νέες ιδέες και έγιναν σύνθετοι μηχανισμοί, τα τμήματα των οποίων βρίσκονται σε συνεχή αλληλεπίδραση. Η κατανόηση της βασικής δομής της μεθόδου Ant Colony Optimization είναι ιδιαίτερα σημαντική καθώς οι πιο επιτυχημένοι αλγόριθμοι της οικογένειας όπως οι Ant System, MAX-MIN Ant System και Ant Colony System δημιουργούνται από αυτή. Τα ποιοτικά αποτελέσματα μιας εφαρμογής ACO στηρίζονται στο σωστό σχεδιασμό του αλγορίθμου και στην αρμονική μεταξύ τους αλληλεπίδραση. Δηλαδή στην επιλογή των τμημάτων του και την ρύθμιση των παραμέτρων των τμημάτων έτσι ώστε να αποδίδει στο υπό έρευνα πρόβλημα.

4.2 Ευριστική Αναζήτηση – Μεταευριστική Αναζήτηση

Ευριστική Αναζήτηση

Ευριστική αναζήτηση (heuristic search), που πολλές φορές αναφέρεται και ως προσεγγιστικός αλγόριθμος, είναι η χρήση ενός πρακτικού κανόνα, μιας απλοποίησης, που έχει ως σκοπό να περιορίσει το χώρο αναζήτησης σε ιδιαίτερα δύσκολα προβλήματα. Οι ευριστικές μέθοδοι, που πολλές φορές αναφέρονται και σαν προσεγγιστικοί αλγόριθμοι, σκοπεύουν στην υπολογιστική απόδοση και στην απλότητα της ιδέας με πιθανό κόστος την ορθότητα και/ή ακρίβεια των λύσεων. Δηλαδή απεμπολώντας την εγγύηση της απόλυτα βέλτιστης λύσης, κερδίζουμε λύσεις που είναι πάρα πολύ κοντά στην θεωρητικά άριστη με τεράστιο κέρδος σε υπολογιστικό χρόνο. Ο όρος προέρχονται από την ελληνική λέξη 'ευρίσκειν'.

Χωρίζονται σε κατασκευαστικές μεθόδους (constructive methods) και μεθόδους τοπικής αναζήτησης (local search methods).

Οι κατασκευαστικές μέθοδοι δημιουργούν λύσεις από μηδενική βάση, προσθέτοντας επαναληπτικά τμήματα της λύσης σε ένα αρχικά κενό σύνολο το οποίο τελικά γίνεται η προτεινόμενη βέλτιστη λύση. Αν και είναι από τις καλύτερες ανάμεσα στις προσεγγιστικές μεθόδους συνήθως η ποιότητα των λύσεων είναι κατώτερες από τις μεθόδους τοπικής αναζήτησης.

Μια μέθοδος τοπικής αναζήτησης ξεκινά από μια αρχική λύση και με επαναλαμβανόμενα προσπαθεί να τη βελτιώσει με τοπικές κινήσεις. Σημαντική για τη λειτουργία μιας τέτοιας μεθόδου είναι ο ορισμός της δομής που έχει η γειτονιά αναζήτησης (neighborhood structure) επί του χώρου αναζήτησης. Πρακτικά ο ορισμός της γειτονιάς κάθε λύσης θέτει το σύνολο των πιθανών λύσεων από το χώρο αναζήτησης. Σε προβλήματα μεγιστοποίησης ή ελαχιστοποίησης ο αλγόριθμος με κινήσεις που συναντώνται στη βιβλιογραφία ως ανάβαση λόφου(hill climbing) ή κλιμακωτή κάθοδος (gradient descent) προσεγγίζει μια καλύτερη λύση μέσα στη γειτονιά της υπάρχουσας λύσης. Αν η νέα λύση είναι καλύτερη από την τρέχουσα, αυτή αντικαθιστάται με την νέα. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να μην βρίσκονται νέες καλύτερες λύσει στη γειτονιά. Το πρόβλημα είναι ο προφανής εγκλωβισμός σε τοπικά βέλτιστα.

Η δημιουργία μικρού αριθμού διαφορετικών λύσεων στις κατασκευαστικές μεθόδους και ο εγκλωβισμός σε τοπικά βέλτιστα για τις τοπικές επαναληπτικές μεθόδους είναι σοβαρότατα μειονεκτήματα. Η ανάγκη για κάτι που θα κατεύθυνε την έρευνα μακριά από αυτά ήταν επιτακτική και η λύση προτάθηκε στη μεταευριστική αναζήτηση.

Μεταευριστική Αναζήτηση

Ο όρος μεταευριστική αναζήτηση (metaheuristic search) προτάθηκε από τον Glover για πρώτη φορά το 1986 στη πρώτη δημοσίευση της μεθόδου Tabu Search. . Οι όροι προέρχονται από τις ελληνικές λέξεις 'μετά' και 'ευρίσκειν'. Σαφέστερα ορίζεται[1] ως μια κυρίαρχη στρατηγική η οποία κατευθύνει και τροποποιεί άλλες ευριστικές μεθόδους για να παράγει λύσεις πέραν αυτών που κανονικά παράγονται κατά την αναζήτηση τοπικού βέλτιστου. Είναι δηλαδή ένα σύνολο αλγοριθμικών βημάτων που μπορούν να χρησιμοποιηθούν για να οριστούν ευριστικές μέθοδοι που εφαρμόζονται σε μεγάλο εύρος προβλημάτων.

[1] Glover, F., & Laguna, M. (1997). Tabu Search. Boston. Kluwer Academic Publishers

Μεταευριστικές μέθοδοι έχουν κυρίως εφαρμοστεί σε προβλήματα που με βάση τη θεωρία της Υπολογιστικής Πολυπλοκότητας κατατάσσονται ως NP-Hard ή NP-Complete. Συνήθης όμως είναι και η χρήση τους σε προβλήματα συνδυαστικής βελτιστοποίησης για τα οποία είναι γνωστό ότι υπάρχει λύση σε πολυωνυμικό χρόνο αλλά δεν είναι πρακτικά απτή.

Παλαιότερες μεταευριστικές μέθοδοι είναι οι Tabu Search, Προσομοιωμένη Ανόπτηση (simulated annealing) και οι γενετικοί αλγόριθμοι (genetic algorithms) οι οποίες αναλύονται σε επόμενο κεφάλαιο.

4. 3 Η Μεταευριστική Μέθοδος Ant Colony Optimization

Η Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO) ορίστηκε από τους Dorigo και Caro[2] ως ένα κοινό γενικό πλαίσιο για τους Αλγόριθμους Μυρμηγκιών. Ως τα πιο σημαντικά στοιχεία ενός αλγόριθμου μυρμηγκιών αναφέρονται η ικανότητα των μυρμηγκιών να κατασκευάσουν μια εφικτή λύση, η θετική και αρνητική ενίσχυση μέσω φερομόνης, διάφορες διεργασίες τύπου daemon actions που δεν μπορούν να κάνουν τα μυρμήγκια όπως κάποια μέθοδος τοπικής έρευνας ή η ανάδειξη του καλύτερου μυρμηγκιού μιας επανάληψης.

Ο βασικός, υψηλού επιπέδου ψευδοκώδικας δίδεται παρακάτω.

[2] Dorigo, M. and Di Caro, G. (1999a). Ant Colony Optimization: A new meta-heuristic. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A., editors, Proceedings of the Congress on Evolutionary Computation, volume 2, pages 1470.1477, May_ower Hotel, Washington D.C., USA. IEEE Press.

Procedure ACO Metaheuristic

while termination criterion not satisfied **do**

begin-Plan Activities

Construct Ant Solutions

Update Pheromones

Daemon Actions % optional

end- Plan Activities

end while

Οι τρεις διαδικασίες Construct Ant Solutions, Update Pheromones, Daemon Actions, είναι ένα γενικό πλαίσιο αναφοράς.

Η διαδικασία Construct Ant Solutions χειρίζεται τον πληθυσμό των μυρμηγκιών τα οποία είτε σε σειρά, δηλαδή ένα-ένα, είτα παράλληλα, δηλαδή όλα μαζί, κινούνται σε κάθε επανάληψη του αλγόριθμου από κόμβο σε κόμβο στο γράφημα του προβλήματος. Η μετακίνηση γίνεται ανάμεσα σε γειτνιάζοντες κόμβους και χρησιμοποιούνται το ίχνος φερομόνης (pheromone trail) και η ευριστική πληροφορία (heuristic information) σε μία στοχαστική διαδικασία σε καθαρά τοπικό επίπεδο. Με αυτόν τον τρόπο βήμα-βήμα τα μυρμήγκια κατασκευάζουν τη διαδρομή-λύση. Μέτα το πέρας της διαδρομής-λύσης ή κατά τη διάρκεια αυτής, γίνεται εκτίμηση αυτής και το αποτέλεσμα χρησιμοποιείται από τη διαδικασία Update Pheromones με σκοπό το προσδιορισμό της ποσότητας φερομόνης που πρέπει να εναποτεθεί στα τόξα της.

Η διαδικασία Update Pheromones έχει ως σκοπό τη τροποποίηση των ιχνών φερομόνης. Το ίχνος φερομόνης σε μια διαδρομή μειώνεται λόγω της εξάτμισης της φερομόνης ή αυξάνεται διότι η διαδρομή χρησιμοποιείται από τα μυρμήγκια. Η συνέπεια του κανόνα ανανέωσης φερομόνης είναι ότι η διαδρομή που προκρίνεται από τα μυρμήγκια ή από τουλάχιστον ένα μυρμήγκι, ενισχύεται με φερομόνη και αυτό αυξάνει την πιθανότητα να επιλεγεί από τα μυρμήγκια σε μελλοντικές επαναλήψεις του αλγορίθμου. Η άλλη πλευρά αυτής της διαδικασίας είναι ότι οι διαδρομές-λύσεις που δεν προκρίθηκαν περνούν στη λήθη. Με αυτό τον τρόπο ξεχνιούνται περιοχές του χώρου αναζήτησης που μπορούν να εγκλωβίσουν τον αλγόριθμο σε τοπικά βέλτιστα και οδηγείται η αναζήτηση σε εξερεύνηση νέων περιοχών.

Η διαδικασία Daemon Actions είναι προαιρετική και εξαρτάται από τη φύση του προβλήματος και την κρίση του ερευνητή. Η εφαρμογή τους συνήθως έχει σκοπό να γίνουν ενέργειες που δεν μπορούν να γίνουν από ένα μυρμήγκι. Μια μέθοδος τοπικής έρευνας για παράδειγμα ή η συλλογή πληροφοριών από το σύνολο της αναζήτησης και όχι μόνο τοπικά μπορούν προσφέρουν στη διαδικασία ανανέωσης της φερομόνης. Το πιο απλό παράδειγμα τέτοιας διαδικασίας είναι ένας κανόνας βάση του οποίου επιλέγεται το καλύτερο μυρμήγκι της επανάληψης του αλγορίθμου και σε αυτό επιτρέπεται η εναπόθεση πρόσθετης φερομόνης στη διαδρομή που χρησιμοποίησε. Πιο εκτεταμένα η διαδικασία Construct Ant Solutions μπορεί να αναλυθεί όπως στον παρακάτω ψευδοκώδικα.

```
Procedure Construct Ant Solutions
while resources <= resources_termination_criterion do
        plan the genesis of a new ant ();
        New_Operational_Ant ();
end while
```

Ως παραδείγματα εφαρμογής της δράσης ενός μυρμηγκιού και του τρόπου χρήσης της φερομόνης δίδονται οι παρακάτω ψευδοκώδικες με δύο διαφορετικές μεθόδους ανανέωσης της φερομόνης. Στο πρώτο παράδειγμα η φερομόνη ανανεώνεται βήμα-βήμα σε πραγματικό χρόνο.

```
Procedure New_Operational_Ant 1
Ant initialization ();
AM ←create and update AntMemory ();
while present state ≠ goal state do
        AR ←learn local ant routing table ();
        CP←calculate transition probabilities (AR,AM, problem constraints);
        next state←apply ant decision rule (CP, problem constraints);
        move to next state (next state);
        if incremental realtime UpdatePheromone then
                deposit pheromone on visited arc ();
                update ant routing table();
        end if
endwhile
terminate all operations ();
```

Μια μικρή παραλλαγή του προηγούμενου παραδείγματος που μας επιτρέπει να δούμε την καλή προσαρμοστικότητα του πλαισίου ACO σε τροποποιήσεις είναι ο επόμενος ψευδοκώδικας όπου η φερομόνη ανανεώνεται μετά το πέρας μια διαδρομής-λύσης.

Procedure New_Operational_Ant 2

Ant initialization ();

AM $\leftarrow$ create and update AntMemory ();

while present state $\neq$ goal state **do**

  AR $\leftarrow$ learn local ant routing table ();

  CP $\leftarrow$ calculate transition probabilities (AR,AM, problem constraints);

  next state $\leftarrow$ apply ant decision rule (CP, problem constraints);

  move to next state (next state);

  if delayed time UpdatePheromone **then**

    evaluate solution();

    deposit pheromone on all visited arc ();

    update ant routing table();

  end if

endwhile

terminate all operations ();

Παρακάτω παρουσιάζεται ένα αναλυτικό παράδειγμα ενός βασικού ψευδοκώδικα ACO με πληθυσμό μυρμηγκιών 'M', τιμές φερομόνης 'τ' αλλά και παράγοντα εξάτμισης φερομόνης 'ρ'.

Procedure ACO Algorithm Basic

Initialize ant population M, $M \rightarrow \{1,2,...,m\}$ and pheromone values τ, $\tau_{ij} \rightarrow \tau_0$

while condition for termination not met **do**

  for each ant $i \in \{1,...m\}$ **do**

```
        initialize solution set S → {0,1,...,n−1}
        ant i constructs solution λ_i
    end for
    for all(i, j) do
        τ_ij → (1 − ρ)·τ_ij
    end for
    compute best solution of iteration λ⁺
    if λ⁺ better than current best λ* then
        set λ* = λ⁺
    end if
    for all(i, j) ∈ λ⁺ do
        τ_ij → (1 − ρ)·τ_ij + ρ·τ_0
    end for
    for all(i, j) ∈ λ* do
        τ_ij → (1 − ρ)·τ_ij + ρ·τ_0
    end for
endwhile
```

Με την έναρξη του αλγορίθμου, αρχικοποιείται και ο πίνακας φερομόνης. Ανάλογα με το υπό έρευνα πρόβλημα ορίζονται το επίπεδο φερομόνης που αρχικοποιείται καθώς και ποιές μεταβλητές. Στο παραπάνω παράδειγμα έχει τροποποιηθεί η ανανέωση της φερομόνης στην ολικά καλύτερη λύση, αλλά και στην καλύτερη της επανάληψης λύση, το τ_0, αρχική τιμή της φερομόνης προστίθεται για πρόκριση των καλών λύσεων.

Είναι σημαντικό ότι το πλαίσιο αλγορίθμων ACO όπως περιγράφεται στον ψευδοκώδικα '**Procedure** ACO Metaheuristic' είναι αρκετά ανοιχτό σε τροποποιήσεις καθώς η διαδικασία '**Plan Activites'** προβλέπει τρείς υπο-διαδικασίες (Construct Ant Solutions, Update Pheromones, Daemon Actions). Σοφά όμως δεν προσδιορίζει με αυστηρότητα και σαφήνεια το πώς εφαρμόζονται και συγχρονίζονται αυτές οι τρείς διαδικασίες. Αυτό είναι κάτι που άπτεται στο πρόβλημα υπό εξέταση και στη κρίση του ερευνητή. Για παράδειγμα σε προβλήματα όπως το TSP, κατά τους Gambardella και Dorigo[3],είναι δυνατό να μειώσουμε το σύνολο των υποψήφιων λύσεων σε ένα σύνολο από καλές επιλεγμένες λύσεις ώστε το μυρμήγκι να επιλέγει μόνο από αυτές. Σκοπός αυτής της κίνησης είναι σε προβλήματα με ιδιαίτερα μεγάλο πλήθος να μειωθεί ο χρόνος που χρειάζεται.

Σε άλλα προβλήματα με βελτιστοποίηση διάσχισης κόμβων, όπως το TSP, είναι σύνηθες να αρχικοποιείται ευριστική πληροφορία[4] ως $\eta_{ij} = \frac{1}{d_{ij}}$, όπου d_{ij} είναι η απόσταση ή 'ορατότητα' μεταξύ των κόμβων i και j. Μια μικρή τιμή για το d_{ij} προσφέρει άμεσο όφελος αν επιλεγεί ο εν λόγω κόμβος καθώς το μήκος της λύσης αυξάνεται κατά μια μικρή τιμή.

Παράγοντας ιδιαίτερης σημασίας είναι η ποσότητα της ευριστικής πληροφορία, καθώς σε κάποιες υλοποιήσεις είναι σταθερή ενώ σε άλλες μεταβάλλεται. Οι Dorigo

---

[3] Gambardella, L. M. and Dorigo, M. (1996). Solving Symmetric and Asymmetric TSPs by Ant Colonies. p 622-627, In ICEC96, Proceedings of the IEEE Conference on Evolutionary Computation, Japan.
[4] Bonabeau, E., Dorigo, M., and Theraulaz, G., (1999). Swarm Intelligence : From natural to artificial systems, Oxford University Press.

και Gambardella σε άλλη εργασία[5] τους προτείνουν μία μέθοδο για την εύρεση του βέλτιστου αριθμού μυρμηγκιών που εξαρτάτε από τη μέση τιμή της φερομόνης αλλά και από το μέγεθος του προβλήματος n. Στη εργασία αυτή ως βέλτιστος αριθμός του πλήθους μυρμηγκιών δίδεται το m=10.

Σήμερα πολλοί αλγόριθμοι αποικίας μυρμηγκιών έχουν εφαρμοστεί σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης. Παρακάτω αναλυτικά παρουσιάζονται οι γνωστότερες και πιο επιτυχημένες υλοποιήσεις αλγορίθμων μυρμηγκιών με σκοπό την βαθύτερη ανάλυση τους.

4.3.1 Simple ACO

Ένα στοιχειώδες πλαίσιο για την κατανόηση των βασικών εννοιών προτάθηκε από τους Dorigo και Stutzle [6], στο οποίο διάφορα χαρακτηριστικά της οικογένειας ACO είτε μειώθηκαν είτε αφαιρέθηκαν τελείως. Ο σκοπός είναι μέσω της απλότητας αυτής της πρότασης να εκτεθούν ξεκάθαρα οι θεμελιώδεις ιδέες και μηχανισμοί που δρουν σε ένα τέτοιο αλγόριθμο. Ο πιθανοκρατικός κανόνας έγινε απλούστερος καθώς η ποσότητα ευρεστικής πληροφορίας εξαλείφτηκε (heuristic information), αλλά και ο κανόνας ανανέωσης (update rule) απλοποιήθηκε έτσι ώστε η βελτίωση στη προτεινόμενη λύση να βασίζεται στο διαφοροποιημένο μήκος μιας διαδρομής.

[5] Dorigo, M., Gambardella, L.M., (1997) Ant Colony System: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation, vol 1, p. 53-66

[6] Dorigo, M. and Stutzle, T. (2001). An experimental study of the simple ant colony optimization algorithm. In Proceedings of the, 2001 WSES International Conference on Evolutionary Computalution Computation. WSES-Press International.

Στη συγκεκριμένη υλοποίηση μια επανάληψη του αλγορίθμου αποτελείται από την μετακίνηση των μυρμηγκιών, την εξάτμιση της φερομόνης και την εναπόθεση της φερομόνης. Οι βασικές ιδέες παρουσιάζονται παρακάτω.

Πιθανοκρατικά κινούμενα προς τα εμπρός μυρμήγκια και κατασκευή λύσης

Στο αλγόριθμο S-ACO, τα μυρμήγκια κινούνται προς τα εμπρός, από τον αρχικό κόμβο (φωλιά) προς τον κόμβο προορισμού (τροφή), και προς τα πίσω, από τον κόμβο προορισμού προς τον αρχικό κόμβο. Μόλις ένα μυρμήγκι φτάσει στον κόμβο προορισμού ξεκινά τη διαδρομή του προς τα πίσω. Θεωρούμε ότι τα μυρμήγκια κινούνται σε ένα γράφημα G=(N,A), όπου οι κόμβοι i και j είναι γειτονικοί ($i, j \in N$) και υπάρχει το τόξο που τους συνδέει ($i, j \in A$). Κινούμενα προς τα εμπρός, επιλέγουν πιθανοκρατικά ποιος θα είναι ο επόμενος κόμβος που θα επιλέξουν από τον κόμβο στον οποίο βρίσκονται. Αυτή η επιλογή επηρεάζεται από τα ίχνη φερομόνης που έχουν εναποτεθεί στο γράφημα από άλλα μυρμήγκια. Σημαντικό είναι ότι κατά την κίνηση προς τα εμπρός δεν εναποθέτουν φερομόνη, και είναι αυτή η επιλογή μαζί με την ντετερμινιστική κίνηση προς τα πίσω που έχουν ως σκοπό να εμποδίσουν τον εγκλωβισμό σε κυκλικές τοπικές λύσεις.

Με αφετηρία τον αρχικό κόμβο, κάθε μυρμήγκι κατασκευάζει μία λύση εφαρμόζοντας επαναληπτικά μια πολιτική λήψης αποφάσεων με σκοπό τη μετάβαση του από κόμβο σε κόμβο.

Η μετάβαση του μυρμηγκιού k που βρίσκεται στο κόμβο i και χρησιμοποιεί το ίχνος φερομόνης τ_{ij} για να υπολογίσει την πιθανότητα να επιλέξει τον κόμβο j γίνεται με βάση την παρακάτω εξίσωση.

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{a}}{\sum_{l \in N_i^k} \tau_{il}^{a}}, & \text{εάν } j \in N_i^k \\ 0, & \text{εάν } j \notin N_i^k \end{cases} \qquad (3.1)$$

Όπου N_i^k η γειτονιά του μυρμηγκιού k που βρίσκεται στο κόμβο i.

Επίσης για τη συγκεκριμένη υλοποίηση θέτουμε ένα σταθερό ποσό φερομόνης σε όλους τους κόμβους κατά την έναρξη. Δηλαδή:

$$\tau_{ij} = 1, \forall (i, j) \in A \qquad (3.2)$$

Η γειτονιά του μυρμηγκιού k αποτελείται από όλους του γειτνιάζοντες κόμβους του i εκτός από τον ακριβώς προηγούμενο κόμβο. Στην περίπτωση που ο κόμβος i είναι αδιέξοδος, το σύνολο N_i^k περιέχει τον ακριβώς προηγούμενο κόμβο.

Εξάτμιση της φερομόνης

Ο κανόνας εξάτμισης της φερομόνης εφαρμόζεται με σκοπό την εξερεύνηση νέων λύσεων έτσι ώστε να αποφευχθεί η πρόωρη σύγκλιση σε μη άριστες λύσεις. Η μείωση των επιπέδων φερομόνης ενισχύει την εξερεύνηση διαφορετικών διαδρομών σε ολόκληρη την περιοχή αναζήτησης. Ο μηχανισμός της εξάτμισης με το να ενισχύει τη λήθη παλαιών μη άριστων λύσεων οδηγεί σε μια βελτιωμένη δομή του προβλήματος που ωφελεί την αποικία. Επίσης είναι εξαιρετικά χρήσιμος καθώς θέτει ένα άνω όριο στη φερομόνη που μπορεί να υπάρχει σε μια διαδρομή.

Αφού το μυρμήγκι k έχει μετακινηθεί στον επόμενο κόμβο, η εξάτμιση της φερομόνης εφαρμόζεται όπως παρακάτω.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in A \tag{3.3}$$

Όπου $\rho \in (0,1]$.

Ντετερμινιστικά κινούμενα προς τα πίσω μυρμήγκια και ανανέωσης φερομόνης με βάση την ποιότητα της λύσης

Στη προς τα πίσω διαδρομή τα μυρμήγκια κινούνται με απόλυτο τρόπο βασισμένα στη μνήμη των κόμβων που απέκτησαν από την κίνησή τους προς τα εμπρός καθώς και το κόστος αυτών εάν το γράφημα είναι σταθμισμένο (weighted). Είναι δυνατόν λοιπόν να υπολογιστεί μια ανανέωση της φερομόνης με βάση το κόστος της κάθε διαδρομής και συνεπώς να προκρίνονται οι ποιοτικότερες λύσεις. Δηλαδή περισσότερη φερομόνη στις συντομότερες διαδρομές, κάτι που ισχύει και σε ορισμένα βιολογικά μυρμήγκια. Σημαντικές διεργασίες σε αυτό το βήμα είναι ότι απαλείφουν τις κυκλικές διαδρομές (loops) και εναποθέτουν φερομόνη στα τόξα που διέσχισαν.

Η απαλοιφή των κυκλικών διαδρομών γίνεται με επαναληπτική έρευνα των δεικτών των κόμβων θέση προς θέση. Για τον κόμβο στη θέση i, ξεκινάμε την έρευνα της διαδρομής από τον τελικό κόμβο μέχρι να τον συναντήσουμε στη θέση j. Η έρευνα σταματά στη θέση i , οπότε πάντα θα είναι $i \leq j$. Εάν ανακαλύψουμε ότι $i > j$, τότε το υποτμήμα της διαδρομής από τη θέση i+1 έως τη θέση j είναι μια κυκλική διαδρομή και πρέπει να απαλοιφή.

Η ανανέωση της φερομόνης κατά τη διαδρομή προς τα πίσω γίνεται όπως παρακάτω.

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^{k} \qquad (3.4)$$

Δηλαδή το μυρμήγκι k κινούμενο στην προς τα πίσω διαδρομή διασχίζει το τόξο (i,j) και αλλάζει την φερομόνη σε αυτό κατά $\Delta\tau^{k}$. Με αυτό τον τρόπο το τόξο (i,j) έχει μεγαλύτερες πιθανότητες να επιλεγεί από τα επόμενα μυρμήγκια στις επόμενες επαναλήψεις του αλγορίθμου. Ο όρος $\Delta\tau^{k}$ έχει σημαντική λειτουργία. Τελέστηκαν υλοποιήσεις του αλγορίθμου με $\Delta\tau^{k}$ σταθερό και $\Delta\tau^{k} = \frac{1}{L^{k}}$, όπου L^{k} το μήκος της διαδρομής του k μυρμηγκιού.

Σε αυτόν τον αλγόριθμο μόνο το διαφοροποιημένο μήκος μιας διαδρομής λειτουργεί υπέρ της επιλογή της συντομότερης διαδρομής. Σε κάθε μονάδα χρόνου τα μυρμήγκια που κινούνται στις συντομότερες διαδρομές εναποθέτουν φερομόνη νωρίτερα σε σχέση με αυτά που κινούνται στις μακρύτερες. Ο παρακάτω πίνακας δείχνει τα αποτελέσματα 100 επαναλήψεων με πείραμα διπλής γέφυρας.

m	1	2	4	8	16	32	64	128	256	512
Χωρίς να ληφθεί υπόψη το μήκος της διαδρομής	50	42	26	29	24	18	3	2	1	0
Λαμβάνοντας υπόψη το μήκος της διαδρομής	18	14	8	0	0	0	0	0	0	0

Πίνακας 3.1

Αποτελέσματα για 100 επαναλήψεις με m τον αριθμό των μυρμηγκιών, με α=2 και ρ=0. Ο αλγόριθμος S-ACO συγκλίνει πολύ πιο γρήγορα προς τη συντομότερη διαδρομή όταν η ποιότητα της λύσης λαμβάνεται υπόψη στην ανανέωση της φερομόνης.

Οι συγγραφείς συμπέραναν ότι για το συγκεκριμένο αλγόριθμο σε ιδιαίτερα μεγάλα προβλήματα βελτιστοποίησης, το διαφοροποιημένο μήκος μιας διαδρομής έχει πεπερασμένη αποτελεσματικότητα, όπως και ο ρυθμός εξάτμισης της φερομόνης. Αν και είναι δυνατόν να ξεπεραστεί αυτό το πρόβλημα χρησιμοποιώντας μεγάλες αποικίες εντόμων, δημιουργούν το πρόβλημα οτι αυξάνουν δυσανάλογα το κόστος σε υπολογιστικό χρόνο.

Επίσης επισήμαναν ότι κανόνας ενημέρωσης της φερομόνης (pheromone update rule), με βάση την ποιότητα των λύσεων, είναι σημαντικός για τη γρήγορη σύγκληση του αλγορίθμου στη βέλτιστη λύση, τονίζοντας όμως το γεγονός ότι μεγάλες τιμές του παράγοντα 'α' τόνιζαν τις επιδράσεις των αρχικών τυχαίων διακυμάνσεων και τελικά οδηγούσαν σε κακή συμπεριφορά του αλγορίθμου.

Τα συμπεράσματα των συγγραφέων δείχνουν το γεγονός ότι η θετική ανάδραση για να λειτουργήσει χρειάζεται ένα συγκεκριμένο μέγεθος προβλήματος, καθώς και το γεγονός ότι σε ένα πρόβλημα συγκεκριμένου μεγέθους δημιουργείται ανάδραση που δρα θετικά στη ποιότητα του της λύσης. Επίσης στο συγκεκριμένο παράδειγμα είτε δίδονται συμπεράσματα με τη δυνατότητα να εξατμίζεται φερομόνη είτε χωρίς και κατά συνέπια ο ρυθμός εξάτμισης της φερομόνης 'ρ' δεν σχολιάζεται επαρκώς.

Η ιδέα της ενημέρωσης της φερομόνης με βάση την ποιότητα της λύσης έχει δειχθεί πειραματικά ότι είναι επιτυχής, όταν το πρόβλημα είναι συνεχούς φύσης, ενώ σε προβλήματα διακριτής φύσης έχει δειχθεί[7] ότι δημιουργούνται προβλήματα με εγκλωβισμό του αλγορίθμου σε τοπικά βέλτιστες λύσεις. Σημαντικό είναι και ότι ο παράγοντας 'α' μπορεί να προκαλέσει λάθη στα αρχικά στάδια του αλγορίθμου που με τη σειρά τους θα διογκωθούν και θα φέρουν μια λύση με τοπικό και όχι ολικό βέλτιστο.

4.3.2 Ant System

Ο Ant System δημοσιεύθηκε από τον Dorigo[8] και σε αυτόν βρίσκουμε τα βασικά χαρακτηριστικά της θετικής ανάδρασης, της κατανεμημένης αρχιτεκτονικής και μια διαδικασία δημιουργίας λύσεων. Αυτά τα χαρακτηρίστηκα ενσωματώθηκαν από τις

[7] Blum, C. and Dorigo, M. (2004). Deception in ant colony optimization. Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004, Proceedings, vol. 3172, p. 118-129. Lecture Notes in Computer Science. Springer.

[8] Dorigo M., Maniezzo V., Colorni A., (1991b) The Ant System: An Autocatalytic Optimizing Process, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy.

προηγούμενες προσπάθειες των ερευνητών που υπάρχουν στους αλγόριθμους Ant Density, Ant Quantity[9] και Ant Cycle[10].

Τα βασικά χαρακτηριστικά του αλγορίθμου είναι:

- Ο Πίνακας Φερομόνης (Pheromone Matrix) που λειτουργεί ως μνήμη του αλγορίθμου και επιτρέπει την έμμεση επικοινωνία μεταξύ των μυρμηγκιών.

- Την πιθανότητα, $p_{ij}^k(t)$ για κάθε μετάβαση του μυρμηγκιού k από μια κορυφή i σε όποια κορυφή j, $j \in N_i^k$ (N_i^k είναι η γειτονιά πιθανών κορυφών από την κορυφή i).

- Ο Κανόνας Ανανέωση Διαδρομής (Trail Update Rule) προσδιορίζει πώς οι νέες λύσεις απεικονίζονται στο πίνακα φερομόνης.

- Την παράμετρο 'ρ' που ορίζεται ως ποσοστό εξάτμισης της φερομόνης στις διαδρομές.

Ο Dorigo διαπίστωσε ότι όταν m=n , δηλαδή ότι πρέπει ο πληθυσμός των μυρμηγκιών και ο αριθμός των πόλεων/κόμβων του υπό έρευνα προβλήματος να είναι ίσος. Μεγάλος πληθυσμός θα οδηγήσει σε τοπικά βέλτιστα, ενώ μικρός εμποδίζει τη δημιουργία στιγμεργετικής επικοινωνίας μεταξύ των μυρμηγκιών.

---

[9] Dorigo M., Maniezzo V., Colorni A., (1991a) Positive feedback as a search strategy.Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[10] Colorni A., Dorigo M., Maniezzo V.,(1992) Distributed optimization by ant colonies. In F.J. Varela & P. Bourgine (Eds.). Proceedings of the first european conference on artificial life. p. 134-142, Cambridge, MA , ,MIT Press.

Επίσης προτείνει ότι η αρχικοποίηση των επιπέδων φερομόνης να τίθεται σε μία τιμή ελαφρά μεγαλύτερη της τιμής που αναμένεται να εναποθέσουν τα μυρμήγκια σε μια επανάληψη του αλγορίθμου. Ένα καλό μέτρο αυτής της τιμής δίδεται από την παρακάτω εξίσωση.

$$\tau_{ij} = \tau_0 = \frac{m}{C^{nn}}, \forall(i, j) \qquad (3.5)$$

όπου: m ο αριθμός των μυρμηγκιών

C^{nn} το μήκος λύσης που προκύπτει από τον αλγόριθμο nearest neighbor

Με τα παραπάνω μέτρα αποφεύγεται το φαινόμενο το αρχικό επίπεδο φερομόνης (τ_0) να είναι πολύ χαμηλό, γεγονός που θα έκανε τις πρώτες επαναλήψεις ιδιαίτερα σημαντικές και θα οδηγούσε σε εγκλωβισμό σε τοπικά βέλτιστα. Από την άλλη πολύ υψηλά επίπεδα αρχικής φερομόνης θα χρειαστούν πολλές επαναλήψεις του αλγορίθμου μέχρι ο παράγοντας εξάτμισης να οδηγήσει τα μυρμήγκια σε εξερεύνηση.

Στη βιβλιογραφία συναντούμε τους όρους J και $tabu_k$. Ισχύει ότι : $tabu_k \subseteq J$ καθώς J είναι το σύνολο των εφικτών και πιθανών επιλογών κορυφών. Το σύνολο $tabu_k$ είναι οι κορυφές που το μυρμήγκι k δεν έχει επισκεφθεί και συνεπώς δεν έχουν συμπεριληφθεί στη μερική λύση. Εναλλακτικά, η λίστα $tabu_k$ εκφράζεται και ως N_i^k, η γειτονιά του κόμβου i για το μυρμήγκι k. Ισχύει ότι $N_i^k \subseteq N_i$, και είναι το σύνολο των κόμβων που γειτνιάζουν με τον κόμβο i και το μυρμήγκι k δεν έχει επισκεφθεί.

Έστω ο πληθυσμός τεχνητών μυρμηγκιών m και ο αριθμός των πόλεων n. Η θέση που επιλέγεται για καθένα μυρμήγκι έχει αφετηρία μια διαφορετική πόλη ή τοποθετούνται τυχαία. Στη λίστα $tabu_k$ στη μνήμη του μυρμηγκιού k η πόλη αυτή είναι πρώτη. Οι πόλεις στο N_i^k επιλέγονται από αυτούς που ανήκουν στο N_i με χρήση της μνήμης του μυρμηγκιού. Σε κάθε επανάληψη t του αλγορίθμου το μυρμήγκι k που βρίσκεται στη πόλη i επιλέγει την επόμενη πόλη j με χρήση του πιθανοκρατικού κανόνα που δίδεται στη παρακάτω εξίσωση.

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}(t)]^{\beta}} & \text{, εάν } j \in N_i^k \\ 0 & \text{, εάν } j \notin N_i^k \end{cases} \tag{3.6}$$

Όπου: $\eta_{ij} = \dfrac{1}{d_{ij}}$ είναι η ποσότητα της ευριστικής πληροφορίας

d_{ij} είναι η απόσταση ή 'ορατότητα' μεταξύ των κόμβων i και j

α,β παράμετροι που σταθμίζουν ευριστική πληροφορία και φερομόνη

N_i^k οι πόλεις που μπορεί να επισκεφθεί το μυρμήγκι

Ο πιθανοκρατικός κανόνας βασίζεται στην πληροφορία από την ένταση της φερομόνης σε μια διαδρομή (τ_{ij}) και την ευρεστική πληροφορία (η_{ij}) που ορίζεται από το χρήστη και επίσης ονομάζεται και ορατότητα. Η ευρεστική πληροφορία είναι μία εκτίμηση της επιτυχίας του αλγορίθμου βασιζόμενη στο πόσο βαθιά ερευνά το χώρο αναζήτησης.

Η τελική επιρροή αυτών των παραμέτρων σταθμίζεται από τις, από τον χρήστη οριζόμενες, παραμέτρους 'α' και 'β'. 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 τιμή θα οδηγήσει στη γρήγορη εμφάνιση στασιμότητας, μιας κατάστασης όπου όλα τα μυρμήγκια κάνουν την ίδια διαδρομή, η οποία είναι κατά πολύ κατώτερη της άριστης.

Ανανέωση Φερομόνης

Οι τρείς αρχικοί αλγόριθμοι δοκιμάστηκαν στο Πρόβλημα του Περιοδεύοντος Πωλητή και διαφέρουν στο χειρισμό της φερομόνης. Ο Ant Density χρησιμοποιεί ένα σταθερό ποσό φερομόνης μετά από ένα βήμα ενός μυρμηγκιού. Ο Ant Quantity χρησιμοποιεί μια ποσότητα ανάλογη της απόστασης μεταξύ δύο πόλεων και ανανεώνει τη φερομόνη κατά $\frac{Q}{c_{ij}}$, όπου Q παράμετρος που τίθεται από τον ερευνητή και c_{ij} το κόστος μετάβασης από την πόλη i στη πόλη j. Ο Ant Cycle χρησιμοποιεί πρώτα μια διαδικασία κατασκευής λύσεων και μετά ανανεώνει τη φερομόνη σε μία διαδρομή με ποσότητα ανάλογη του μήκους (L) της διαδρομής ίση με $\frac{Q}{L^k}$,όπου L^k το μήκος της διαδρομής που κατασκεύασε το k μυρμήγκι. Ο Ant System υιοθετεί διαδικασία ανανέωσης της διαδρομής παρόμοια του Ant Cycle.

Μετά το πέρας όλων των διαδρομών των μυρμηγκιών στη επανάληψη t, εναποθέτουν φερομόνη όπως παρακάτω.

$$\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t), \quad \forall (i,j) \in L \tag{3.7}$$

Όπου ρ, $0 \le \rho < 1$ ο ρυθμός εξάτμισης της φερομόνης.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{1}{C^{k}}, & \text{εάν } (i,j) \in T^{k} \\ 0, & \text{εάν } (i,j) \notin T^{k} \end{cases} \tag{3.8}$$

Όπου: T^{k} η διαδρομή (tour) του μυρμηγκιού k

C^{k} το μήκος ή κόστος της διαδρομής T^{k}

Ο παράγοντας ρ περιορίζει τη συσσώρευση φερομόνης στις διαδρομές και επιτρέπει στον αλγόριθμο να ξεχνά τις μη βέλτιστες διαδρομές. Γενικά όμως, όσα περισσότερα μυρμήγκια επιλέξουν ένα τόξο ij τόσο περισσότερη φερομόνη θα λάβει κατά τη διαδικασία ανανέωσης με αποτέλεσμα στις επόμενες επαναλήψεις να είναι πιθανότερο να επιλεγεί ξανά.

Εφαρμογές Ant System

Ο αλγόριθμος Ant System έχει δοκιμαστεί και στο πρόβλημα χρωματισμού γράφων[11] (graph coloring problem) από τους Costa και Hertz (1997) όπου οι ερευνητές δημιούργησαν μνήμη για την ανάθεση των μέσων στις ανάγκες, αλλά και μνήμη για την διάταξη των αντικειμένων. Σε αυτή την εργασία δίδεται ένα γενικό πλαίσιο για την εφαρμογή του αλγορίθμου σε Προβλήματα Ανάθεσης (assignment type problems) και πάνω σε αυτό εφαρμόζουν τον Ant System.

Οι ερευνητές έδειξαν ότι είναι εφικτό να εφαρμοστεί ο Ant System σε προβλήματα ανάθεσης και τόνισαν ότι το μέγεθος της αποικίας (δηλαδή το πλήθος των μυρμηγκιών) είναι σημαντικό, καθώς σε μια μη παράλληλη μηχανή αποτελεί το ρυθμιστικό παράγοντα μεταξύ του χρόνου που απαιτείται και της ποιότητας της λύσης. Η εργασία δείχνει ότι πέρα κάποιου σημείου, μεγαλύτεροι αριθμοί μυρμηγκιών δεν οδηγούν σε καλύτερες λύσεις.

Ο Ant System δοκιμάζεται στο Τετραγωνικό Πρόβλημα Εκχώρησης (Quadratic Assignment Problem) από τους Maniezzo και Colorni[12] (1999) με ελαφρές τροποποιήσεις που δείχνονται στην παρακάτω εξίσωση αλλά και με την ενσωμάτωση μεθόδου τοπικής έρευνας.

[11] Costa, D. and Hertz, A. (1997). Ants can colour graphs. Journal of the Operational Research Society, issue 48, p. 295-305.
[12] Maniezzo, V. and Colorni, A. (1999) The ant system applied to the quadratic assignment problem. IEEE Transactions on Knowledge and Data Engineering, vol 11, issue 5 p. 769-778.

$$
p_{ij}^{k}(t) = \begin{cases} \dfrac{\alpha \cdot \tau_{ij}(t-1) + (1-\alpha)\cdot \eta_{ij}}{\sum_{r \in allow_{k}} (\alpha \cdot \tau_{ir}(t) + (1-\alpha)\cdot \eta_{ir})} & , \text{εάν } j \in N_{i}^{k} \\ 0 & , \text{εάν } j \notin N_{i}^{k} \end{cases} \tag{3.9}
$$

Οι τιμές των παραμέτρων που δοκιμάστηκαν δίδονται στο παρακάτω πίνακα.

Παράμετρος	**Προτεινόμενες Τιμές**	**Τιμή με τα Καλύτερα Αποτελέσματα**
α	[0,3][0,5][0,9]	[0,5]
ρ	[0,7],[0,9], [0,95],[0,99]	[0,9]

Υψηλές τιμές του 'α' οδήγησαν σε στασιμότητα του αλγορίθμου, ενώ δεν δίδεται σχόλιο για τις χαμηλές τιμές. Αναφέρεται επίσης, ότι χαμηλές τιμές του 'ρ' μείωσαν την απόδοση του αλγορίθμου καθώς χρειάστηκε περισσότερος χρόνος να βρεθεί ποιοτική λύση. Επίσης παρατηρήθηκε ότι ο αριθμός των μυρμηγκιών δεν επηρέαζε σημαντικά την απόδοση του αλγορίθμου.

Ο αλγόριθμος Ant System, συνδυασμένος με μέθοδο τοπικής έρευνας, συγκρινόμενος με το αλγόριθμο Grasp[13] παρήγαγε αποτελέσματα σχεδόν όμοιας ποιότητας. Τα παρεμφερή αποτελέσματα οφείλονται στην ομοιότητα των αλγορίθμων. Ο Grasp δημιουργεί λύσεις με τη διαδικασία Greedy Randomised Construction βάση του ορίου 'α' (threshold α) και της τυχαίας επιλογής. Στις παλαιότερες εφαρμογές του αλγορίθμου το όριο το έθετε ο αυθαίρετα ο ερευνητής αλλά στις νεότερες ρυθμίζεται

[13] Resende, M. and Ribeiro, C. (2003). Greedy randomized adaptive search procedures. Glover, F. and Kochenberger, G. (eds), Handbook of Metaheuristics, p219.249. Kluwer Academic Publishers.

βάση του υπάρχοντος περιβάλλοντος. Αλλά οι ερευνητές κατέληξαν ότι ο Ant System είναι καλύτερος και ότι αυτό οφείλεται στον πίνακα φερομόνης που αποτελεί τη μνήμη του αλγορίθμου. Βασική διαφορά ανάμεσα τους είναι ο τρόπος χρήσης της διαδικασίας μνήμης καθώς ο Grasp καταγράφει τη καλύτερη έως τώρα λύση αλλά δεν υπάρχει η δυνατότητα χρήσης πληροφοριών από προηγούμενες επαναλήψεις.

4.3.3 Elitist Ant System

Η πρώτη αλλαγή του Ant System είναι ο Elitist Ant System και προτάθηκε από τον Dorigo[14]. Η ενσωμάτωση της ιδέας των εκλεκτών μυρμηγκιών (elitist ants) έχει σκοπό να δοθεί πρόσθετη ενίσχυση στη βέλτιστη διαδρομή που έχει βρεθεί μέχρι εκείνη τη στιγμή, την T^{bs} (best tour so far). Αυτή η πρόσθετη ενίσχυση γίνεται βάση του $\frac{e}{C^{bs}}$, όπου e παράγοντας που σταθμίζει το βάρος που θέλουμε να έχει η T^{bs} και C^{bs} το κόστος/μήκος αυτής.

Η ανανέωσης φερομόνης γίνεται όπως παρακάτω.

$$\tau_{ij}(t+1) \leftarrow (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) + e\Delta\tau_{ij}^{bs}(t) \qquad (3.10)$$

με

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{1}{C^{k}}, \text{ εάν } (i,j) \in T^{k} \\ 0 \quad, \text{ εάν } (i,j) \notin T^{k} \end{cases} \qquad (3.11)$$

---

[14] Dorigo M., Maniezzo V., Colorni A., (1996) Ant System: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics. Part B, 26 (1) p. 29–41.

Όπου: T^k η διαδρομή (tour) του μυρμηγκιού k

C^k το μήκος/κόστος της διαδρομής T^k

και

$$\Delta \tau_{ij}^{bs}(t) = \begin{cases} \dfrac{1}{C^{bs}}, \text{ εάν } (i,j) \in T^{bs} \\ 0 \quad , \text{ εάν } (i,j) \notin T^{bs} \end{cases} \tag{3.12}$$

Όπου: T^{bs}, η καλύτερη διαδρομή μέχρι εκείνη τη στιγμή (best tour so far)

C^{bs} ,το μήκος/κόστος της διαδρομής T^{bs}

Κατά τον Dorigo η ελιτίστικη στρατηγική σε συνδυασμό με σχετικό με το πρόβλημα παράγοντά 'e' , δίνουν καλύτερα αποτελέσματα και σε λιγότερες επαναλήψεις από τον Ant System.

4.2.4 MAX MIN Ant System

Καθώς οι πρώτοι αλγόριθμοι μυρμηγκιών γίνονταν ευρύτερα γνωστοί, οι ερευνητές παρατήρησαν κάποια σημαντικά χαρακτηριστικά. Περισσότερη έμφαση στην εκμετάλλευση είδη υπαρχουσών λύσεων αντί για εξερεύνηση νέων οδηγούσε σε καλύτερες και γρηγορότερες λύσεις. Αναδεικνύονταν το μεγάλο πρόβλημα των ελιτίστικων μεθόδων οι οποίες οδηγούσαν σε σχετικά σύντομη χρονικά σύγκλιση προς μια μη βέλτιστη λύση. Το πρόβλημα της στασιμότητας άρχισε να γίνεται περισσότερο κατανοητό και ως λύση σε αυτό προτάθηκε[15] ο παράγοντας διακλάδωσης κόμβων (node branching factor λ). Ο παράγοντας διακλάδωσης κόμβων 'λ' ενός κόμβου/πόλης είναι ο αριθμός των ακμών που συνδέουν τον κόμβο/πόλη με τους υπόλοιπους κόμβους και που έχουν ένα ποσό φερομόνης πάνω από ένα όριο. Παρατηρήθηκε ότι οι καλύτερες λύσεις βρίσκονταν όταν ο 'λ' παράγοντας δημιουργίας κλάδων ήταν κοντά στο 1, αλλά όταν ο 'λ' ήταν ίσος με 1 η εύρεση νέων λύσεων σχεδόν μηδενιζόταν.

Ο Max-Min Ant System προτάθηκε από τους Stuzle και Hoos[16] και δοκιμάστηκε με επιτυχία απέναντι στον Ant System στο TSP και το QAP. Ο νέος αλγόριθμος εκμεταλλεύεται σε μεγάλο βαθμό τις καλύτερες διαδρομές που έχουν βρεθεί καθώς μόνο το μυρμήγκι με την καλύτερη διαδρομή της επανάληψης ή αυτό με τη γενικά καλύτερη έως τώρα λύση επιτρέπεται να εναποθέσει φερομόνη. Σοβαρή παρενέργεια μιας τέτοιας επιλογής είναι η στασιμότητα (stagnation) λύσεων που είναι η κατάσταση κατά την οποία όλα τα μυρμήγκια ακολουθούν την ίδια διαδρομή. Σε αυτή την περίπτωση η υπερβολική συσσώρευση φερομόνης σε μία διαδρομή τα εγκλωβίζει. Συνεπώς για να αποφευχθεί αυτή η υπερβολική συσσώρευση φερομόνης τίθενται άνω και κάτω όρια σε αυτή. Παρακάτω παρουσιάζεται πιο αναλυτικά ο MAX-MIN Ant System που πρότεινε τις εξής νέες ιδέες:

- Μόνο ένα άριστο μυρμήγκι στο τέλος της κάθε επανάληψης του αλγορίθμου μπορούσε να ανανεώσει τον πίνακα φερομόνης.

[15] Bonabeau E., Dorigo M., Theraulaz G., Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, Oxford, 1999.
[16] Stutzle, T. and Hoos, H. (1997). The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In Back T., Michalewize Z, Yao X. (Eds.), Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC 97), p. 309-314, IEEE Press

- Άνω και Κάτω Όρια (τ_{max} τ_{min}) τίθεντο για την ένταση της φερομόνης στις διαδρομές.

- Οι διαδρομές αρχικοποιούνται με αυστηρά ορισμένο τρόπο.

Ανανέωση Φερομόνης

Ο νέος κανόνας ανανέωσης ορίζει ότι μόνο ένα μυρμήγκι σε κάθε επανάληψη ανανεώνει το πίνακα φερομόνης. Αυτό είναι είτε αυτό με την ολικά βέλτιστη διαδρομή (global best ant), είτε αυτό με την καλύτερη της επανάληψης (iteration best ant). Η επιλογή επηρεάζει τη συμπεριφορά του αλγορίθμου. Όταν η ανανέωση της φερομόνης γίνεται από το ολικά καλύτερο μυρμήγκι, η έρευνα συγκλίνει γρήγορα προς την T^{bs} ενώ αν γίνεται από το καλύτερο μυρμήγκι της επανάληψης αυτό δεν συμβαίνει τόσο γρήγορα καθώς περισσότερες διαδρομές ενισχύονται.

Ο νέος κανόνας ανανέωσης παρουσιάζεται παρακάτω.

$$\tau_{ij}(t+1) \leftarrow \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{best}(t) \qquad (3.13)$$

με ρ: $0 \le \rho < 1$ ο ρυθμός εξάτμισης της φερομόνης

$$\Delta\tau_{ij}^{best}(t) = \begin{cases} \dfrac{1}{C^{bs}}, \text{ εάν } (i,j) \in T^{bs} \\ 0 \quad, \text{ εάν } (i,j) \notin T^{bs} \end{cases} \qquad (3.14)$$

Όπου: T^{bs} η καλύτερη διαδρομή μέχρι εκείνη τη στιγμή (best tour so far)

C^{bs} το μήκος/κόστος της διαδρομής T^{bs}

Εναλλακτικά, αν χρησιμοποιηθεί το καλύτερο μυρμήγκι της επανάληψης είναι:

$$\Delta\tau_{ij}^{best}(t)=\begin{cases}\dfrac{1}{C^{ib}}, \text{ εάν } (i,j)\in T^{ib}\\ 0 \quad, \text{ εάν } (i,j)\notin T^{ib}\end{cases} \tag{3.15}$$

Όπου: T^{ib} ,η καλύτερη διαδρομή της επανάληψης (iteration best)

C^{ib} ,το μήκος/κόστος της διαδρομής T^{ib}

Το ποίο μυρμήγκι επιλέγεται για την ανανέωση είναι στην ευχέρειά μας καθώς μπορούμε να επιλέξουμε κάθε φορά αυτό με την ολική βέλτιστη λύση ή αυτό με την ολική βέλτιστη λύση κάθε k επαναλήψεις. Ο Stutzle, αναφέρει ότι για μικρά προβλήματα περιοδεύοντος πωλητή (TSP) καλύτερα και πιο γρήγορα αποτελέσματα δίνει η επιλογή του καλύτερου της επανάληψης μυρμηγκιού, ενώ για μεγάλα προβλήματα καλύτερη απόδοση επιτυγχάνεται με την επιλογή του ολικά καλύτερου μυρμηγκιού.

Τα Όρια της Φερομόνης

Τα άνω τ_{max} και κάτω τ_{min} όρια της φερομόνης τίθενται με σκοπό να αποφευχθεί η στασιμότητα του αλγορίθμου. Με το άνω όριο εξασφαλίζεται ότι καμία διαδρομή δε θα συγκεντρώσει υπερβολικά μεγάλα ποσά φερομόνης εγκλωβίζοντας το χώρο αναζήτησης των μυρμηγκιών. Το κάτω όριο εγγυείται ότι καμία διαδρομή δε θα μείνει χωρίς φερομόνη και συνεπώς σχεδόν μηδενική πιθανότητα επιλογής.

Το άνω όριο βρίσκεται αν όπου C^{best} θέσουμε το μήκος/κόστος της θεωρητικά απόλυτα βέλτιστης διαδρομής και δίδεται από την παρακάτω εξίσωση:

$$\tau_{max}(t) = \frac{1}{\rho \cdot C^{best}} \tag{3.16}$$

Στην πράξη εφαρμόζεται ότι $C^{best} = C^{bs}$. Δηλαδή ως θεωρητικά απόλυτα βέλτιστη διαδρομή θεωρείται η καλύτερη που έχει ανακαλυφθεί μέχρι εκείνη τη στιγμή. Συνεπώς κάθε φορά που ανακαλύπτουμε μια νέα ολικά βέλτιστη διαδρομή, ανανεώνουμε και το τ_{max}.

Για δείξουμε[17] ότι η εξίσωση (3.16) ισχύει αρκεί να λάβουμε t επαναλήψεις του αλγορίθμου. Για $t = 0,1,2,...,t$ έχουμε βάση του κανόνα ανανέωση της φερομόνης (3.13) ότι :

Για $t = 1$, $\quad \tau_{ij}(1) = (1-\rho)\tau_{ij}(0) + \dfrac{1}{C^{best}}$

Για $t = 2$, $\quad \tau_{ij}(2) = (1-\rho)\tau_{ij}(1) + \dfrac{1}{C^{best}} = (1-\rho)^2\tau_{ij}(0) + \dfrac{(1-\rho)}{C^{best}} + \dfrac{1}{C^{best}}$

[17] Φαινέκος Γ., (2001), Η Μέθοδος Βελτιστοποίησης Με Αποικίες Μυρμηγκιών: Εφαρμογή σε Διακριτά Και Συνεχή Προβλήματα , Διπλωματική εργασία, Τμήμα Μηχανολόγων Μηχανικών, Εθνικό Μετσόβιο Πολυτεχνείο.

Για $t = 3$, $\tau_{ij}(3) = (1-\rho)\tau_{ij}(2) + \frac{1}{C^{best}} = (1-\rho)^3 \tau_{ij}(0) + \frac{(1-\rho)^2}{C^{best}} + \frac{(1-\rho)}{C^{best}} + \frac{1}{C^{best}}$

$$\vdots$$

Για $t = t$, $\tau_{ij}(t) = (1-\rho)^t \tau_{ij}(0) + \frac{1}{C^{best}} \sum_{i=0}^{t-1} (1-\rho)^i \quad (3,17)$

Συνεπώς το όριο για $t \to \infty$ θα είναι:

$$\lim_{t\to\infty} \tau_{ij}(t) = \lim_{t\to\infty} (1-\rho)^t \tau_{ij}(0) + \frac{1}{C^{best}} \lim_{t\to\infty} \sum_{i=0}^{t-1} (1-\rho)^i \quad (3,18)$$

Άλλα είναι $\rho < 1$, συνεπώς:

$$\lim_{t\to\infty} (1-\rho)^t \tau_{ij}(0) = 0 \quad (3,19)$$

και

$$\lim_{t\to\infty} \sum_{i=0}^{t-1} (1-\rho)^i = \frac{1}{\rho} \quad (3,20)$$

Οπότε η εξίσωση (3,18) γίνεται:

$$\lim_{t\to\infty} \tau_{ij}(t) = \frac{1}{C^{best}} \cdot \frac{1}{\rho} \quad (3,21)$$

Η έννοια της σύγκλισης στον MAX-MIN Ant System έχει επιτευχθεί όταν για κάθε κόμβο υπάρχει μια σύνδεση με άλλο κόμβο όπου τη τιμή της φερομόνης είναι τ_{max} και όλες οι υπόλοιπες συνδέσεις έχουν φερομόνη ίση με τ_{min}. Σε αυτή τη περίπτωση η βέλτιστη διαδρομή έχει κατασκευαστεί με πιθανότητα σημαντικά μεγαλύτερη του μηδέν. Έστω ότι η πιθανότητα επιλογής της σωστής διαδρομής, σε όλους τους κόμβους είναι ίδια και ίση με p_{dec}. Αν η πιθανότητα επιλογή της βέλτιστης λύσης είναι p_{best} τότε :

$$p_{best} = p_1 \cdot p_2 \cdot p_3 \cdots p_n \tag{3,22}$$

και

$$p_1 = p_2 = p_3 = \ldots = p_n = p_{dec} \tag{3,23}$$

Άρα:

$$p_{best} = p_{dec}^{n} \Leftrightarrow p_{dec} = \sqrt[n]{p_{best}} \tag{3,24}$$

Για $\eta_{ij} = 1$ και για πλήθος τόξων $\frac{n}{2}$ που συνδέουν τον κόμβο που βρίσκεται, το μυρμήγκι, με τους υπόλοιπους κόμβους, ο πιθανοκρατικός κανόνας μετάβασης γίνεται:

$$p_{dec} = \frac{\tau_{max}}{\tau_{max} + \left(\frac{n}{2} - 1\right) \cdot \tau_{min}} \tag{3,25}$$

Άρα το τ_{max} μπορεί να εκφραστεί ως:

$$\tau_{max} = p_{dec} \cdot \left(\tau_{max} + \left(\frac{n}{2} - 1 \right) \cdot \tau_{min} \right)$$

$$\tau_{max} = p_{dec} \cdot \tau_{max} + p_{dec} \cdot \left(\frac{n}{2} - 1 \right) \cdot \tau_{min}$$

Συνεπώς ,

$$\tau_{min} = \frac{\tau_{max} \cdot (1 - p_{dec})}{\left(\frac{n}{2} - 1 \right) \cdot p_{dec}} \tag{3,26}$$

ή ισοδύναμα

$$\tau_{min} = \frac{\tau_{max} \cdot (1 - \sqrt[n]{p_{best}})}{\left(\frac{n}{2} - 1 \right) \cdot \sqrt[n]{p_{best}}} \tag{3,27}$$

Ο Stutzle[18] προτείνει ως πρακτικό κανόνα για το κάτω όριο της φερομόνης τον $\tau_{min} = \frac{\tau_{max}}{\alpha}$, όπου 'α' παράγοντας που ορίζεται από τον ερευνητή. Τα αποτελέσματά του επίσης τονίζουν ότι το κάτω όριο φερομόνης είναι σημαντικότερο του άνω για την αποφυγή της στασιμότητας, χωρίς να ξεχνά να αναφέρει τη χρησιμότητα του άνω

[18] Stutzle T. & Hoos H.H., (2000) MAX-MIN Ant System. Future Generation of Computer Systems, vol 16 issue 8, p. 889-914

ορίου σε περιπτώσεις που χρειάζεται να επαναρχικοποιήσουμε (reinitialize) τον αλγόριθμο.

Αρχικοποίηση Διαδρομών

Η αρχικοποίηση των επιπέδων φερομόνης είναι ίση με το τ_{max}, και αυτό δίνει μεγάλη σημασία στο 'ρ' (ρυθμός εξάτμισης φερομόνης) διότι οι μη αποδοτικές διαδρομές, λόγω του 'ρ', εξαφανίζονται και ο αλγόριθμος συγκλίνει ταχύτερα προς τη βέλτιστη λύση. Δηλαδή όταν $\tau_0 = \tau_{max}$ κατά τις πρώτες επαναλήψεις γίνεται καλύτερη εξερεύνηση του χώρου αναζήτησης διότι η διαφορά μεταξύ των καλών και των κακών λύσεων είναι μικρή. Στην περίπτωση που το αρχικό επίπεδο φερομόνης είναι χαμηλό, η διαφορά μεταξύ των καλών και των κακών λύσεων δυσκολότερο να ξεπεραστεί από τα μυρμήγκια τα οποία εκμεταλλεύονται τοπικές λύσεις.

Σε ορισμένα προβλήματα με ιδιαίτερα μεγάλο αριθμό επαναλήψεων πραγματοποιείται επαναρχικοποίηση (reinitialize) της φερομόνης με σκοπό την καλύτερη εξερεύνηση του χώρου αναζήτησης. Αυτό συμβαίνει συνήθως όταν ο αλγόριθμος έχει σχεδόν συγκλίνει. Σε σχέση με το τ_{max}, αυξάνουμε τη φερομόνη στις διαδρομές που σχεδόν δεν έχουν. Ο μηχανισμός παρουσιάζεται παρακάτω.

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \delta \cdot (\tau_{max}(t) - \tau_{ij}(t)) \qquad (3.28)$$

όπου $0 \le \delta \le 1$, παράμετρος που καθορίζεται από τον ερευνητή.

Η τιμή του 'δ' προσδιορίζει και την επιρροή της διαδικασίας στο μηχανισμό εξερεύνησης.

Εφαρμογές MAX-MIN Ant System

Ο αλγόριθμος χρησιμοποιήθηκε από τον Stutzle[19] στο πρόβλημα ροής καταστήματος (Flow Shop Problem) με δύο αλγόριθμους τοπικής έρευνας. Ο πρώτος ήταν ένας έρευνας αντιμετάθεσης (swapping search) όπου για μια περιοχή έρευνας $(n-1)^2$ χρειάζεται $O(n^2m)$ χρόνο υπολογισμού. Ο δεύτερος ήταν μία στρατηγική πρώτης βελτίωσης (fisrt-improvement strategy) και έδινε καλύτερα αποτελέσματα από την στρατηγική καλύτερης βελτίωσης (best-improvement strategy). Σκοπός ήταν όσον το δυνατό λιγότερες επαναλήψεις και οι παράμετροι τέθηκαν με 'ρ' ίσο με 0.75 , 'p' ίσο με $\frac{n-4}{n}$ και $\tau_{min} = \frac{\tau_{max}}{5}$. Τα αποτελέσματα ήταν εξαιρετικά καθώς ο αλγόριθμος απέδωσε καλύτερα από τις μεθόδους της Προσομοιωμένης Ανόπτησης (Simulated Annealing) και της Πολλαπλής Καθόδου (Multiple Descent).

Επίσης από τον Socha[20] στο πρόβλημα Πανεπιστημιακού Προγράμματος (University Timetable Problem) όπου απέδωσε καλύτερα από ένα αλγόριθμο τοπικής έρευνας που ξεκινούσε από τυχαία σημεία στο πρόβλημα. Η έρευνα χρησιμοποιεί ως καλύτερη αναπαράσταση λύσεων τη μέθοδο 'συμβάν σε δωμάτιο με συγκεκριμένο χρόνο' (event to room-specific time slot) και δεν χρησιμοποιεί ευριστική πληροφορία όταν συνδυάζεται με μέθοδο τοπικής έρευνας όποτε στηρίζεται αποκλειστικά στο πίνακα φερομόνης.

---

[19] Stutzle T.(1998). An ant approach to the flow shop problem. In Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing, p.1560--1564, Verlag Mainz, Aachen, Germany

[20] Socha, K., Sampels, M., and Manfrin, M. (2003). Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art. In Proceedings ofEvoCOP 2003 – 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization, volume 2611 of Lecture Notes in Computer Science, p. 334.345. Springer-Verlag.

Ακόμα ο αλγόριθμος έχει χρησιμοποιηθεί σε καινοτόμα προβλήματα όπως στο 2D HP Protein Folding Problem[21] και στο High Level Synthesis[22]. Η απόδοση του αλγορίθμου ήταν καλή, ειδικά καθώς αυξανόταν το μέγεθος του προβλήματος, και σε σχέση με τους Γενετικούς Αλγορίθμους δεν ήταν αναγκαίο να οριστεί τελεστής διασταύρωσης (cross-over operator).

Η εργασία των Stutzle και Dorigo[23] προσφέρει μια απόδειξη σύγκλισης για τους αλγορίθμους μυρμηγκιών MMAS. Σύμφωνα με αυτή, όταν τείνει στο άπειρο ο χρόνος επαναλήψεων, ο αλγόριθμος θα συγκλίνει στη βέλτιστη λύση, όπου ο πίνακας φερομόνης θα έχει τις εγγραφές του ίσες με τ_{min} εκτός από αυτές που αποτελούν μέρος της λύσης και θα έχουν αξία ίση με τ_{max}.

4.2.5 Ant-Q

Οι Gambardella και Dorigo [24] πρότειναν βελτιώσεις στο Ant System με ιδέες εμπνευσμένες από τη ενισχυτική μάθηση (reinforcement learning). Πιο συγκεκριμένα, ο Watkins[25] εισήγαγε τον αλγόριθμο Q-learning ο οποίος λειτουργεί μαθαίνοντας μια συνάρτηση, της οποίας η τιμή σχετίζει την προσδοκώμενη ωφέλεια

[21] Shmygelska, A., Hern'andez, R. A., and Hoos, H. H. (2002). An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. Proceedings of the Third International Workshop, ANTS 2002. Proceedings in Lecture Notes in Computer Science (LNCS) series, Vol. 2463, Springer-Verlag.
[22] Kemprasit, R. and Chongstitvatana, P. (2004). High-level synthesis by dynamic ant. International Journal of Intelligent Systems, vol 19 issue 1-2,p.25.38.
[23] Stutzle, T. and Dorigo, M. (2002). A short convergence proof for a class of aco algorithms. In IEEE Transactions on Evolutionary Computation, vol 6,issue 4 p 358.365.
[24]Gambardella L.M., Dorigo M., (1995) Ant-Q: A reinforcement learning approach to the travelling salesman problem. Proceedings of the Twelfth International Conference on Machine Learning , p. 252-260 , Morgan-Kaufman
[25] Watkins,C,1989.Learning from Delayed Rewards,Thesis,University of Cambidge,England.

από ένα φάσμα δυνατόν δράσεων ,σε μια δεδομένη κατάσταση, χωρίς γνώση του περιβάλλοντος.

Πιο αναλυτικά, για κάθε κατάσταση s, από το σύνολό των καταστάσεων S, και για κάθε δράση a, από το σύνολο των δράσεων A, μπορούμε να υπολογίσουμε την ανανέωση της προσδοκώμενης ωφέλειας από:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + a_t(s_t,a_t)[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)]$$

Όπου: r_t μια πραγματική τιμή τη στιγμή t

$a_t(s,a)$ ρυθμοί μάθησης τέτοιο ώστε $0 \leq a_t(s,a) \leq 1$

γ ρυθμιστικός παράγοντας τέτοιος ώστε $0 \leq \gamma \leq 1$

Παρακάτω παρουσιάζεται ο ψευδοκώδικας του αλγορίθμου:

Q Learning Algorithm

1. Set parameter γ, and environment reward matrix R

2. Initialize matrix Q as zero matrix

3. For each episode:

Select random initial state

Do while not reach goal state

Select one among all possible actions for the current state

Using this possible action, consider to go to the next state

Get maximum Q value of this next state based on all possible actions

Compute:

Q(state, action) = R(state, action) + $\gamma \cdot$ max[Q(next state, all actions)]

Set the next state as the current state

End Do

End For

Η παράμετρος 'γ' που λαμβάνει τιμές στο [0,1] ρυθμίζει τη συμπεριφορά του αλγορίθμου. Όταν το 'γ' είναι κοντά στο 0 ο αλγόριθμος θα τείνει προς μια άπληστη συμπεριφορά καθώς θα προτιμά την άμεση ωφέλεια μιας κίνησης. Αντίστοιχα, όταν το 'γ' είναι κοντά στο 1, ο αλγόριθμος θα υπολογίσει με μεγαλύτερο βάρος την μελλοντική ωφέλεια μιας κίνησης, καθυστερώντας ένα πιθανό βραχυπρόθεσμο όφελος. Πρέπει να σημειώσουμε τον πίνακα ωφέλειας 'R' οποίος είναι αρκετά όμοιος με τον πίνακα φερομόνης, αλλά και ότι και οι δύο αλγόριθμοι προσπαθούν να μάθουν για το περιβάλλον από τις δράσεις τους.

Στον Ant-Q ο πίνακας φερομόνης αντικαθίσταται από τις τιμές Ant-Q, τις AQ_{ij}, που υπολογίζονται με παρόμοιο τρόπο με τις τιμές Q στη ενισχυτική μάθηση (reinforcement learning) και αναπαριστούν την ωφελιμότητα μιας κίνησης. Οι τιμές AQ ανανεώνονται όπως παρακάτω:

$$AQ(r,s) \leftarrow (1-\rho)\cdot AQ(r,s)+\rho\cdot(\Delta AQ(r,s)+\gamma\cdot\max_{z\in J_k} AQ(s,z))$$

Όπου : J_k η λίστα με τις πόλεις που δεν έχει επισκεφθεί το μυρμήγκι k

γ ρυθμιστικός παράγοντας τέτοιος ώστε $0 \le \gamma \le 1$

Σημαντικές καινοτομίες ήταν η εισαγωγή της ιδέας δύο ανανεώσεων, μια τοπικής και μίας ολικής, αλλά και του παράγοντα διακλάδωσης 'λ' (λ branching factor).

Η χρήση της ολικής διαδικασίας ανανέωσης φερομόνης οδηγούσε στην πρόωρη σύγκλιση του αλγορίθμου και ως λύση για να γίνει διεύρυνση στον πίνακα φερομόνης προτάθηκε τη τοπική διαδικασία ανανέωσης με παρόμοιους κανόνες από τον Ant System.

Ο παράγοντας διακλάδωσης 'λ'(λ branching factor), μια συγκεκριμένη χρονική στιγμή, υπολογίζει προσεγγιστικά το μέγεθος του χώρου αναζήτησης και παρουσιάζεται παρακάτω.

Έστω $AQ_{\max}(r,s)$ και $AQ_{\min}(r,s)$ οι μέγιστες και ελάχιστες τιμές AQ των ακμών που εξέρχονται του κόμβου 'r'. Είναι $\delta_r = AQ_{\max}(r,s) - AQ_{\min}(r,s)$. Δοθείσης μια τιμής 'λ', $0 \leq \lambda \leq 1$, η παράμετρος διακλάδωσης 'λ' του κόμβου 'r' είναι ίση με τον αριθμό των ακμών που εξέρχονται του 'r' και έχουν τιμές AQ μεγαλύτερες από $\lambda \cdot \delta_r + AQ_{\min}(r,s)$.

Η παραπάνω εξίσωση μπορεί να εκφραστεί ο πιο απλά ως:

$$\lambda\,(t) \leftarrow \lambda \cdot (\tau_{\max}(i,j) - \tau_{\min}(i,j)) + \tau_{\min}(i,j)$$

Με $\tau_{\max}, \tau_{\min}$ τη μέγιστη και ελάχιστη τιμή του πίνακα φερομόνης και την παράμετρο 'λ' να λαμβάνει τιμές στο [0,1] και να ορίζεται από τον ερευνητή.

Σε μία γραμμή του πίνακα φερομόνης, το πλήθος των τιμών που είναι μεγαλύτερες του 'λ' δίνουν τον παράγοντα διακλάδωσης για τον κόμβο που απεικονίζεται στη σειρά του πίνακα. Αν πολλαπλασιάσουμε για όλους τους κόμβους του γραφήματος μπορούμε να υπολογίσουμε το χώρο αναζήτησης. Καθώς ο 'λ' μειώνεται, οδηγούμαστε σε ένα μονότονα φθίνοντα χώρο αναζήτησης που βοηθά τον αλγόριθμο. Συνήθως στις πειραματικές διατάξεις προτιμούνται χαμηλές τιμές του 'λ' όπως 0,04 , 0,06 , 0,08 και 0,1.

Επίσης στο πιθανοκρατικό κανόνα μετάβασης του Ant-Q προτάθηκε το q_0 για να εξισορροπηθεί η εκμετάλλευση και η εξερεύνηση νέων λύσεων. Το q_0 , $0 \le q_0 \le 1$, είναι παράμετρος τέτοια ώστε μεγαλύτερες τιμές της οδηγούν σε μικρότερη πιθανότητα να συμβεί μια τυχαία επιλογή κόμβου.

Ένα μυρμήγκι k επιλέγει να μετακινηθεί από την πόλη r στη πόλη s βάση της όπως παρακάτω:

$$s = \begin{cases} \arg\max\limits_{u \in J_k(r)} \{[AQ(r,u)]^{\delta} \cdot [\mathrm{HE}(r,u)]^{\beta}\}, & \text{if } q \le q_0 \\ S & \text{, otherwise} \end{cases}$$

Όπου :

δ, β παράμετροι που σταθμίζουν τη σημασία των τιμών AQ και της ευριστικής πληροφορίας,

q είναι μια ομοιόμορφα κατανεμημένη τυχαία μεταβλητή

S τυχαία μεταβλητή που επιλέγεται από την κατανομή της πιθανότητας που ορίζεται από συνάρτηση του AQ(r,u) και HE(r,u) με $u \in J_k(r)$.

Ο Ant-Q έχει δοκιμαστεί με επιτυχία στο πρόβλημα της τετραγωνικής ανάθεσης (QAP) και στο πρόβλημα του περιοδεύοντος πωλητή (TSP) και απέδωσε καλύτερα από τον Ant System. Είναι 'πρόγονος' του Ant Colony System καθώς η σπουδαιότερη διαφορά τους είναι ο ορισμός του τ_0. Αυτή η παράμετρος στον Ant-Q λαμβάνει την τιμή $\tau_0 = \gamma \max_{j \in N_i^k} \{\tau_{ij}\}$, όπου 'γ' παράμετρος που λαμβάνει τιμές στο [0,1] και το μέγιστο ίχνος φερομόνης από τα ίχνη φερομόνης που συνδέουν τον κόμβο i με τον κόμβο j που επιλέγει το μυρμήγκι k. Παρατηρήθηκε ότι για μικρές τιμές του τ_0 ο αλγόριθμος δεν απέδιδε ικανοποιητικά και εγκαταλείφθηκε για τον απλούστερο και πιο αποδοτικό Ant Colony System.

4.3.6 Ant Colony System

Το 1996 οι Dorigo[26] και Gambardella δημοσιεύουν τον αλγόριθμο Ant Colony System (ACS) και τον εφαρμόζουν σε διακριτά προβλήματα βελτιστοποίησης. Τα αποτελέσματα του Ant Colony System είναι καλύτερα από του Ant System. Ο εν λόγω αλγόριθμος αποτελεί εξέλιξη που προέκυψε από ιδέες του Ant-Q και του Ant System που αποσκοπούν να εξισορροπήσουν την ανακάλυψη (exploration) νέων λύσεων, την εκμετάλλευση (exploitation) των ήδη υπαρχουσών λύσεων και την αποφυγή πρόωρης σύγκλισης σε τοπικά βέλτιστα.

Οι νέες ιδέες είναι : ένας νέος πιθανοκρατικός κανόνας μετάβασης με κριτήριο το q_0 και δύο διαδικασίες ανανέωσης της φερομόνης, μια τοπική και μία ολική.

[26] Dorigo, M., Gambardella, L.M., (1997) Ant Colony System: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation, 1 (1), p. 53-66

Ο νέος κανόνας μετάβασης για το μυρμήγκι k που βρίσκεται στη πόλη i και επιλέγει την πόλη j ορίζεται ως :

$$j = \begin{cases} \max_{l \in N_i^k} \{\tau_{il}(t) \cdot [\eta_{il}(t)]^{\beta}\}, & \text{αν } q \leq q_0 \\ J & , \text{αν } q > q_0 \end{cases} \quad (3.29)$$

Όπου : q μια τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο [0,1]

q_0 παράμετρος που ορίζεται από τον ερευνητή στο [0,1], $0 \leq q_0 \leq 1$

J κόμβος που επιλέγεται με βάση τον πιθανοκρατικό κανόνα επιλογής αλλά με α=1

Ο πιθανοκρατικός κανόνας επιλογής ορίζεται ως:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}(t)]^{\beta}} & , \text{εάν } j \in N_i^k \\ 0 & , \text{εάν } j \notin N_i^k \end{cases} \quad (3.30)$$

Όπου: η_{ij} η ποσότητα της ευριστικής πληροφορίας

α,β παράμετροι που σταθμίζουν ευριστική πληροφορία και φερομόνη (α=1)

N_i^k οι πόλεις που μπορεί να επισκεφθεί το μυρμήγκι

Όταν $q \leq q_0$, επιλέγεται η εκμετάλλευση της είδη υπάρχουσας λύσης-γνώσης στον εξερευνημένο χώρο, που έχει δημιουργηθεί από την συλλογική συσσώρευση φερομόνης στις διαδρομές. Δηλαδή επιλέγουν τοπικά βέλτιστες λύσεις με συνέπεια να μην ευρεθεί μια νέα πιθανά ολικά βέλτιστη λύση.

Όταν $q>q_0$, επιλέγεται η εξερεύνηση του χώρου αναζήτησης και γίνεται προσπάθεια να επιλεγεί η καλύτερη τη συγκεκριμένη χρονική στιγμή λύση με βάση μια πιθανότητα. Επίσης, η μέθοδος θεωρεί πιθανό το ενδεχόμενο κάποιο μυρμήγκι να αποκλίνει σημαντικά (δηλαδή να 'χαθεί') και να ανακαλύψει μια νέα καλύτερη διαδρομή.Με πιθανότητα q_0 επιλέγεται ο καλύτερος επόμενος κόμβος με βάση τη 'μνήμη' του αλγορίθμου που αποτυπώνεται στα ίχνη φερομόνης. Ο ορισμός του q_0 είναι καθοριστικός για την εξερευνητική συμπεριφορά του αλγορίθμου, καθώς επιλέγουμε αν θα εστιάσει σε υπάρχουσες βέλτιστες διαδρομές ή θα εξερευνήσει νέες διαδρομές.

Ανανέωση της φερομόνης

Σε αντίθεση με τον Ant System, όπου όλα τα μυρμήγκια εναποθέτουν φερομόνη, στον Ant Colony System η εναπόθεση φερομόνης γίνεται μόνο από το μυρμήγκι που έχει ανακαλύψει την καλύτερη διαδρομή μέχρι εκείνη τη χρονική στιγμή. Η διαδικασία αυτή έχει ως σκοπό να προκαλέσει συμπεριφορά εξερεύνησης στον αλγόριθμο με στόχο νέα ολικά βέλτιστη λύση.

Η ολική ανανέωση της φερομόνης (global pheromone update) γίνεται όπως παρακάτω:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\cdot\Delta\tau_{ij}^{best}(t) \qquad (3,31)$$

Με ρ: $0 \leq \rho < 1$ ο ρυθμός εξάτμισης της φερομόνης,

$$\Delta\tau_{ij}^{best}(t) = \begin{cases} \frac{1}{C^{bs}}, \text{ εάν } (i,j) \in T^{bs} \\ 0 \quad, \text{ εάν } (i,j) \notin T^{bs} \end{cases} \tag{3.32}$$

Όπου: T^{bs} η καλύτερη διαδρομή μέχρι εκείνη τη στιγμή (best tour so far)

C^{bs} το μήκος/κόστος της διαδρομής T^{bs}

Η εναπόθεση φερομόνης γίνεται μόνο στις ακμές που ανήκουν στη T^{bs} και όχι σε όλες όπως συμβαίνει στον Ant System. Συνέπεια αυτής της επιλογής είναι ότι η υπολογιστική πολυπλοκότητα από $O(n^2)$ μειώνεται σε $O(n)$, με n ίσο με τον αριθμό των κόμβων/πόλεων στο υπό έρευνα πρόβλημα. Ο νέος κανόνας εναπόθεσης φερομόνης δημιουργεί ένα νέο ίχνος φερομόνης που είναι ο σταθμισμένος μέσος, με ρυθμιστή το 'ρ', του παλαιού ίχνους και του νέου που εναποτίθεται.

Με το να επιτρέπεται η ανανέωση τη φερομόνης μόνο στο μυρμήγκι με την έως τώρα βέλτιστη διαδρομή ελλοχεύει ο κίνδυνος να εγκλωβιστεί ο αλγόριθμος σε μία μόνο διαδρομή, η οποία είναι κάποιο τοπικό βέλτιστο. Η τοπική ανανέωση της φερομόνης, έχει ως σκοπό να απομακρύνει ένα τέτοιο κίνδυνο με ένα μηχανισμό αρνητικής ανάδρασης (negative feedback). Έτσι για το κάθε απλό μυρμήγκι k που μεταβαίνει από την πόλη i στη πόλη j αφαιρείται από το μονοπάτι που τις συνδέει ένα ποσό φερομόνης βάση του ρυθμού εξάτμισης 'ρ'. Σημειώνεται ότι ο ρυθμός εξάτμισής για την ολική και την τοπική διαδικασία ανανέωσης φερομόνης είναι δυνατό να διαφέρει.

Η τοπική ανανέωση της φερομόνης (local pheromone update) γίνεται με βάση την παρακάτω εξίσωση:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\tau_0 \qquad (3,33)$$

Όπου : ρ ο ρυθμός εξάτμισης της φερομόνης, $0 \le \rho < 1$

τ_0 η ποσότητα της φερομόνης κατά την αρχικοποίηση

Οι Dorigo και Gambardella πειραματικά προσδιορίζουν για το τοπικό 'ρ' ως καλή τιμή ρ=0.1 και για το τ_0, $\tau_0 = \frac{1}{nC^{nn}}$, όπου n είναι ο αριθμός των πόλεων στο υπό εξέταση πρόβλημα και C^{nn} το κόστος/μήκος της διαδρομής που προκύπτει από τον αλγόριθμο nearest neighbor heuristic. Επίσης σημειώνουν ότι η δημιουργία λύσεων στον Ant System μπορεί να γίνει σειραϊκά (sequentially) ή παράλληλα (parallel) δίχως να επηρεαστεί ο αλγόριθμός. Αυτή η επιλογή όμως επηρεάζει τον Ant Colony System καθώς υπάρχει η τοπική διαδικασία ανανέωσης φερομόνης. Στις περισσότερες υλοποιήσεις η κατασκευή λύσεων γίνεται με παράλληλη κίνηση των μυρμηγκιών δίχως να υπάρχουν σημαντικά πειραματικά αποτελέσματα που να προτείνουν τη μια έναντι της άλλης μεθόδου.

Επίσης, σε μια προσπαθεί να γίνει ταχύτερος ο αλγόριθμος σε προβλήματα όπως το Asymetric TSP, καθώς πρέπει να εξεταστούν μία προς μία οι διαδρομές, προτάθηκε η χρήση λίστας υποψήφιων πόλεων (candidate list). Η ταξινόμηση γίνεται με βάση κάποιο ευριστικό κριτήριο. Για παράδειγμα στο TSP ένα τέτοιο κριτήριο μπορεί να είναι η απόσταση. Έτσι, σε αυτή τη λίστα περιλαμβάνετε ένα νούμερο από τις πιο σημαντικές πόλεις ταξινομημένες από την κοντινότερη στη πιο μακρινή. Έτσι πρέπει πρώτα να εξετάσουν τα μυρμήγκια τις πόλεις που ανήκουν στη λίστα με τις υποψήφιες πόλεις. Όταν οι πόλεις που ανήκουν στην λίστα υποψήφιων πόλεων (candidate list) ανήκουν και στη λίστα με τις πόλεις που έχει επισκεφθεί το μυρμήγκι (Tabu list), μπορεί το μυρμήγκι να κατευθυνθεί προς νέες πόλεις.

Ακόμα έχει προταθεί[27] να γίνεται ανανέωση της φερομόνης από τα μυρμήγκια με τις δύο καλύτερες διαδρομές καθώς και αφαίρεση φερομόνης από τις χειρότερες. Ως ένα μέτρο για περεταίρω βελτίωση της απόδοσης του αλγορίθμου θεωρείται και ο συνδυασμός του με μία μέθοδο τοπικής έρευνας. Ο ACS εφαρμόζεται από τους Bautista και Pereira[28] στο Assembly Line Balancing Problem με μέθοδο τοπικής έρευνας και αποδίδει πολύ καλά. Επίσης, ο Silva[29] χρησιμοποιεί ACS για βελτιστοποίηση διαδικασιών Logistic (Logistic Process Optimizing) όπου οι λύσεις που προτείνονται από τον αλγόριθμο είναι οι καλύτερες. Ο ACS δίχως μέθοδο τοπικής έρευνας, έχει εφαρμοστεί σε προβλήματα διακριτής βελτιστοποίησης, το πρόβλημα της τετραγωνικής ανάθεσης[30] (quadratic assignment problem), το πρόβλημα του περιοδεύοντος πωλητή[31] (traveling salesman problem) και στο προβλήματα σειραϊκής διάταξης[32] (sequential ordering problems).

[27] Bonabeau E., Dorigo M., Theraulaz G., Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, Oxford, 1999.
[28] Bautista J., and Pereira J.(2002) Ant algorithms for assembly line balancing. Proceedings of the Third International Workshop on Ant Algorithms. Lecture Notes In Computer Science; Vol. 2463 p 65-75 , Springer Verlag
[29] Silva C. A., Runkler T. A., da Costa Sousa J. M., and Palm R. (2002) Ant Colonies as Logistic Processes Optimizers. Proceedings of the Third International Workshop on Ant Algorithms. Lecture Notes In Computer Science; Vol. 2463 p 76 - 87. Springer Verlag
[30] Maniezzo V. and Colorni A. (1999). The ant system applied to the quadratic assignment problem. IEEE Transactions on Knowledge and Data Engineering, vol 11(5), p769-778.
[31] Dorigo, M., Gambardella, L.M., (1997) Ant Colony System: A cooperative learning approach to the travelling salesman problem, IEEE Transactions on Evolutionary Computation, 1 (1), p. 53-66
[32] Gambardella L. M. and Dorigo M. (1997). HAS-SOP: Hybrid ant system for the sequential ordering problem. Technical Report IDSIA-11-97, IDSIA, Lugano, Switzerland.

4.3.7 Hypercube ACO

Ο Blum[33] έχει προτείνει το σύστημα Υπερκύβου Αλγόριθμου Βελτιστοποίησης Μυρμηγκιών (Hypercube ACO) όπου ενσωματώνονται ιδέες από το Δυαδικό Προγραμματισμό Ακεραίων (Binary Integer Programming) με σκοπό τη διακύμανση της φερομόνης στο [0,1]. Η έμπνευση προήλθε από τη μαθηματική έκφραση πολλών προβλημάτων συνδυαστικής βελτιστοποίησης, όπου οι λύσεις μπορούν να απεικονιστούν σαν δυαδικά ανύσματα. Στο ΔΠΑ τα τμήματα που αποτελούν μέρος της λύσης λαμβάνουν τιμή ίση με 1 και αντίστοιχα αυτά που δεν αποτελούν μέρος της λύση τιμή 0. Μια λύση στο πρόβλημα είναι μια κορυφή του n-διάστατου (n-dimensional) κύβου, όπου n ο αριθμός των μεταβλητών απόφασης. Για να δημιουργηθεί μια λύση με πραγματική τιμή με τη μορφή ενός κυρτού συνδυασμού δυαδικών ανυσμάτων, πρέπει να χαλαρώσουν οι περιορισμοί του προβλήματος για να λάβουν οι μεταβλητές απόφασης τιμές στο $[0,1]$. Το σύνολο εφικτών λύσεων $\tilde{S}$ είναι όλα τα ανύσματα $\vec{v} \in \mathrm{IR}^n$ που είναι κυρτοί συνδυασμοί δυαδικών ανυσμάτων $\vec{x} \in \mathrm{IB}^n$.

$$\vec{v} \in \tilde{S} \Leftrightarrow \vec{v} = \sum_{i \in \mathrm{IB}^n} \gamma_i \cdot \vec{x}_i$$

και

$$\gamma_i \in [0,1], \sum \gamma_i = 1$$

[33] Blum, C., Roli, A., and Dorigo, M. (2001). HC-ACO: The hyper-cube framework for Ant Colony Optimization. In Proceedings of MIC'2001 – Meta–heuristics International Conference, volume 2, p. 399-403, Porto, Portugal.

Η σχέση αποσαφηνίζεται περισσότερο όταν ομαλοποιούμε τις τιμές της φερομόνης ώστε να βρίσκονται στο $[0,1]$, όπου το άνυσμα τη φερομόνης είναι $\vec{\tau}=(\tau_1,\tau_2,\tau_3,...,\tau_n)$ και είναι σημείο στο $\tilde{S}$. Στη περίπτωση που το $\vec{\tau}$ είναι δυαδικό άνυσμα είναι και λύση του προβλήματος.

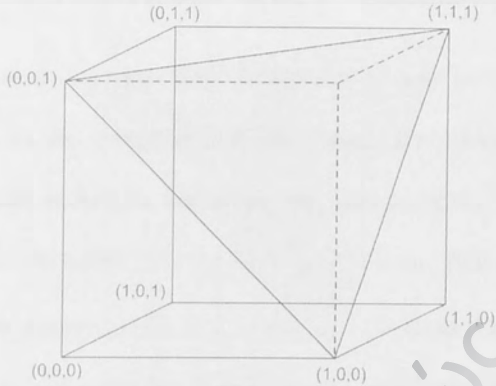
Ανανέωση Φερομόνης

Στον αλγόριθμο Hypercube ACO τα ίχνη της φερομόνης πρέπει να κινούνται αυστηρά στο διάστημα $[0,1]$ και αυτό επιτυγχάνεται με την τροποποίηση του κανόνα ανανέωσης της φερομόνης του Ant System. Η νέα μορφή του κανόνα δίδεται παρακάτω.

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

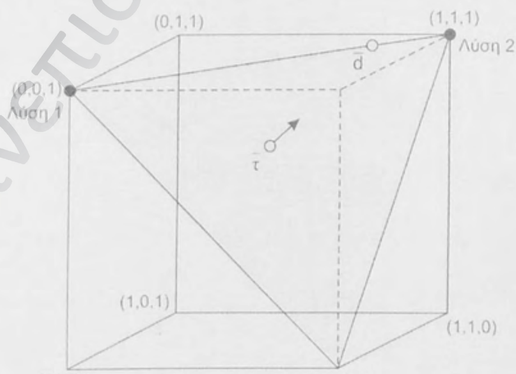
Όπου: $$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1/C^{k}}{\sum_{h=1}^{m} 1/C^{h}}, & \text{αν το τόξο (i,j) χρησιμοποιήθει απο το k μυρμήγκι} \\ 0, & \text{αν το τόξο (i,j) δεν χρησιμοποιήθει απο το k μυρμήγκι} \end{cases}$$

Σκοπός του νέου κανόνα είναι να διασφαλίσει ότι η φερομόνη στο σύστημα παραμένει πάντα μικρότερη του 1. Η λειτουργία του κανόνα δίδεται στο σχήμα 3.2 όπου το νέο άνυσμα της φερομόνης ερμηνεύεται ως μια μετακίνηση από το παλαιό άνυσμα προς το νέο άνυσμα, που προκύπτει από το σταθμισμένο μέσο των λύσεων που χρησιμοποιήθηκαν στον κανόνα ανανέωσης.



Σχήμα 3.1

Στο σχήμα 3.1 δίδεται το σύνολο των εφικτών λύσεων S που είναι τα ανύσματα (0,0,1), (1,0,0) και (1,1,1). Το άνυσμα της φερομόνης $\vec{\tau}$ κινείται στη γκρίζα σκιαγραφημένη περιοχή που είναι η $\tilde{S}$.



Σχήμα 3.2

Οι λύσεις που κατασκευάστηκαν από τα μυρμήγκια είναι οι λύσεις 1 και 2 που απεικονίζονται με τα δύο μαύρα σημεία στο σχήμα 3.2. Είναι οι λύσεις που θα χρησιμοποιηθούν στη διαδικασία ανανέωσης της φερομόνης και είναι οι (0,0,1) και (1,1,1). Θεωρούμαι ότι η λύση (1,1,1) είναι καλύτερη της (0,0,1) και για αυτό το λόγω το $\vec{\tau}$ είναι πιο κοντά στο (1,1,1). Συνεπώς το $\vec{\tau}$ θα κινηθεί προς το $\vec{d}$ το οποίο είναι ο σταθμισμένος μέσος των δύο λύσεων και για αυτό βρίσκεται πάνω στο τμήμα που τις συνδέει.

Σε αυτό τον αλγόριθμο εισάγονται οι έννοιες της ολικής ελκυστικότητας και της ολικής συχνότητας. Η ολική ελκυστικότητα είναι μια τιμή φερομόνης, για κάθε τόξο που αναπαριστά την ελκυστικότητα και η ολική συχνότητα είναι ένας αριθμός που δείχνει πόσες φορές το εν λόγω τόξο έχει χρησιμοποιηθεί. Με αυτά τα χαρακτηριστικά ο αλγόριθμος αυτός μπορεί να εφαρμοστεί σε μεγάλους χώρους αναζήτησης. Χρησιμοποιήθηκε στο Edge-Weighted k Cardinality Problem[34] και στο First Order Shop Scheduling Problem.[35]

[34] Blum, C. and Sampels, M. (2002b). When model bias is stronger than selection pressure. Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN 02), vol. 2439, in Lecture Notes in Computer Science p. 893-902, Springer Verlag.

[35] Blum, C. and Sampels, M. (2002a). Ant colony optimization for FOP shop scheduling: A case study on different pheromone representations. , Proceedings of the 2002 Congress on Evolutionary Computation (CEC 02), p. 1558-1563, Piscataway, NJ, IEEE Press.

4.2.8 P-ACO

Οι Guntsch και Middendorf εισήγαγαν πρώτοι την ιδέα για ένα αλγόριθμο βασισμένο πάνω στον πληθυσμό των μυρμηγκιών. Ο αλγόριθμο Population ACO εφαρμόστηκε σε στατικές[36] αλλά και δυναμικές[37] εκδόσεις των προβλημάτων του περιοδεύοντος πωλητή (TSP) αλλά και του προβλήματος της τετραγωνικής ανάθεσης (QAP).

Η δημιουργία λύσεων στον αλγόριθμο P-ACO είναι όπως στον απλό ACO και οι κύριες διαφορές τους εντοπίζονται στη χρήση της μνήμης, δηλαδή της φερομόνης, και την διαδικασίας ανανέωσης.

Ο P-ACO αρχικοποιεί την φερομόνη για κάθε κόμβο και τη θέτει ίση με μια αρχική τιμή τ_0 η οποία ορίζεται από το χρήστη, δηλαδή $\forall i, j : \tau_{ij} = \tau_0$, και ένα καινό πληθυσμό $P = \emptyset$. Στο τέλος μιας επανάληψης, η καλύτερη λύση π+ θα προστεθεί στο πληθυσμό και εάν έχει γίνει υπέρβαση του ορίου πληθυσμού k , μια λύση $\sigma \in P$ αφαιρείται από τον πληθυσμό. Ο αλγόριθμος χρησιμοποιεί τη θετική ανάδραση αλλά και την αρνητική ανάδραση καθώς όταν προστεθεί η λύση π+ , μέρη που ανήκουν σε αυτή αυξάνουν τη φερομόνη τους κατά την τιμή Δ, και αντίστοιχα όσα ανήκουν στη λύση σ που αφαιρείται, μειώνουν τη φερομόνη τους κατά την τιμή Δ. Η προαναφερθήσα τιμή Δ είναι σταθερή Ο παρακάτω ψευδοκώδικας εκθέτει τους μηχανισμούς του αλγορίθμου.

[36] Guntsch M. Middendorf M. (2002 b) A population based approach for ACO. in Applications of Evolutionary Computing. Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim. vol. 2279. p. 71-80. Springer.
[37] Guntsch M. Middendorf M. (2002 a) Applying Population Based ACO to Dynamic Optimization Problems. in Ant Algorithms. Third International Workshop, ANTS 2002, Proceedings. p. 111-122. vol. 2463 Lecture Notes in Computer Science. Springer-Verlag.

Population ACO Algorithm

initialize pheromone values $\tau_{ij} \to \tau_0$

initialize population $P \to \varnothing$

repeat

  for each ant $i \in \{1, \ldots m\}$ **do**

    initialize selection set $S \to \{0, 1, \ldots n-1\}$

    let ant i construct solution π_i

  end for

  determine best solution of iteration π^+

  add π^+ to population $P \to P \cup \{\pi^+\}$

  for all $(i, j) \in \pi^+$ **do**

    $\tau_{ij} \to \tau_{ij} + \Delta$

  end for

  if $|P| > k$ **then**

    remove solution σ from population $P \to P \setminus \sigma$

    for all $(i, j) \in \sigma$ **do**

      $\tau_{ij} \to \tau_{ij} - \Delta$

    end for

  end if

until condition for termination met

Ως στρατηγικές ανανέωσης του πληθυσμού προτείνονται η ανανέωση βάση της ηλικίας , βάση της πιθανότητας μιας ποιοτικής λύσης, ενός συνδυασμού και των δύο και τέλος με βάση μια καθαρά ελιτίστικη μέθοδο.

Η στρατηγική ανανέωσης βάση της ηλικίας είναι η πιο εύκολη γιατί μετά από k επαναλήψεις απλά αφαιρούμε από τον πληθυσμό την αρχαιότερη λύση. Στη στρατηγική με βάση την πιθανότητα μιας ποιοτικής λύσης, επιλέγεται από τον πληθυσμό P μία λύση με μια πιθανότητα που στηρίζεται στη ποιότητά της.

Πιο συγκεκριμένα, έστω ότι $f(\pi)$ απεικονίζει μια τιμή που σχετίζεται με τη λύση 'π' και ότι χαμηλότερες τιμές της $f(\pi)$ δείχνουν καλύτερη ποιότητα στη λύση.

Με $P=\{\pi_1,....,\pi_{k+1}\}$, ορίζεται η κατανομή των πιθανοτήτων επί των λύσεων π_i, $i=1,...,k+1$ ως:

$$p_i = \frac{x_i}{\sum_{j=1}^{k+1} x_j}$$

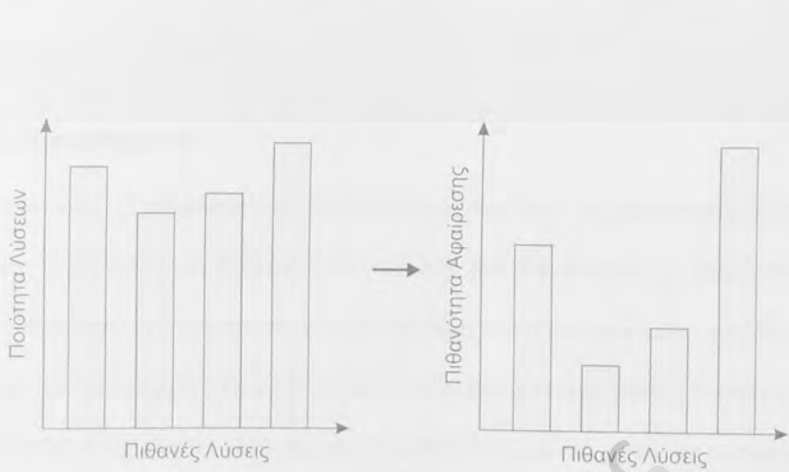
με

$$x_i = f(\pi_i) - \min f(\pi_j) + avg(\pi)$$

και

$$avg(\pi) = \frac{1}{k+1}\sum_{j=1}^{k+1} f(\pi_j) - \min_{j=1,...,k+1} f(\pi_j)$$

Το παρακάτω σχήμα δείχνει τη μετάβαση από την ποιότητα μιας λύσης στην πιθανότητα αφαίρεσης της από τον πληθυσμό.



Σχήμα 3.3

Στο σχήμα 3.3 παρουσιάζεται το πώς σχετίζεται η ποιότητα μιας πιθανής λύση με τη πιθανότητα να αφαιρεθεί από τον πληθυσμό. Είναι εφικτό να δημιουργηθεί μια στρατηγική ανανέωσης από το συνδυασμό των στρατηγικών ηλικίας και πιθανότητας ποιότητας, όπου πρώτα εφαρμόζεται η αξιολόγηση βάση της ποιότητας μιας λύσης και μετά προστίθεται στο πληθυσμό. Η ελιτίστικη ανανέωση του πληθυσμού βασίζεται στην ολικά βέλτιστη λύση π_e. Αυτή η λύση δεν υπόκειται στις προαναφερθείσες στρατηγικές και μπορεί να αφαιρεθεί από τον πληθυσμό μόνο από μια άλλη ολικά βέλτιστη. Το ελίτ μυρμήγκι έχει το βάρος $w_e \in [0,1]$ και ανανεώνει τη φερομόνη με τιμή ίση με $\Delta_e = w_e \cdot \tau_{\max}$ ενώ τα απλά k μυρμήγκια με $\Delta = (1 - w_e) \cdot \tau_{\max} / k$. Ο αλγόριθμος P-ACO είναι ιδιαίτερα ευπροσάρμοστος και σε δυναμικά προβλήματα καθώς η πληροφορία για το ίχνος φερομόνης διατηρείται στον πληθυσμό της αποικίας. Συνεπώς είναι σχετικά εύκολο να διορθωθούν οι λύσεις που έχει προτείνει η αποικία και με βάση τις νέες λύσεις να δημιουργηθεί το νέο ίχνος φερομόνης. Οι Guntsch και Middendorf δοκιμάζουν την απόδοση του αλγορίθμου στο πρόβλημα του περιοδεύοντος πωλητή (TSP) και της τετραγωνικής ανάθεσης (QAP) και τον βρίσκουν ότι αποδίδει καλά αποτελέσματα.

4.3 Συμπεράσματα

Οι ευριστικές και μεταευριστικές μέθοδοι έχουν σαν στόχο τη προσέγγιση βέλτιστων λύσεων σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης με σημαντικότατα οφέλη στον υπολογιστικό χρόνο. Το οικοσύστημα των μεταευριστικών μεθόδων είναι πλούσιο και πολυσχιδές. Το κοινό σημείο όλων των μεθόδων είναι η προσπάθεια να αποφευχθεί η δημιουργία λύσεων χαμηλής ποιότητας με τη χρήση μηχανισμών που επεκτείνουν τις δυνατότητες αλγορίθμων που έχουν σχεδιαστεί για τη λύση συγκεκριμένων προβλημάτων είτε αυτοί είναι κατασκευαστικοί αλγόριθμοι είτε αλγόριθμοι τοπικής έρευνας. Τα στοιχεία διαφοροποίησης των μεταευριστικών μεθόδων έχουν να κάνουν με τις τεχνικές που χρησιμοποιούν για να αποφύγουν τον εγκλωβισμό σε τοπικά βέλτιστα και τον τρόπο που αυτές υλοποιούνται κατά τη δημιουργία των λύσεων.

Τα κύρια σημεία ενός αλγορίθμου μυρμηγκιών είναι : ο πίνακας φερομόνης, ο πιθανοκρατικός κανόνας κατασκευής λύσεων, ο κανόνας ανανέωσης της φερομόνης, η ευριστική πληροφορία και η προαιρετική ενσωμάτωση daemon action συνήθως με τη μορφή μιας μεθόδου τοπικής έρευνας.

Ο πίνακας φερομόνης είναι η μνήμη του αλγορίθμου και είναι η απεικόνιση της συλλογικής εμπειρίας της αποικίας για το περιβάλλον της. Αποτελείται από τα στοιχεία που χρειάζονται για την κατασκευή των λύσεων. Για παράδειγμα στο πρόβλημα του περιοδεύοντος πωλητή είναι ένας πίνακας του οποίου οι γραμμές και στήλες αντιστοιχούν στις συνδέσεις μεταξύ των πόλεων και οι τιμές του σχετίζονται με την πιθανότητα επιλογής κάθε πόλης σαν επόμενο κόμβο στη λύση.

Ο πιθανοκρατικός κανόνας κατασκευής λύσεων συνδυάζει την ευριστική πληροφορία και τις πληροφορίες που απεικονίζονται στο πίνακα φερομόνης για να επιλεγεί ο επόμενος κόμβος που θα προστεθεί στη τρέχουσα λύση. Ο κανόνας ανανέωσης της φερομόνης είναι ο τρόπος με τον οποίο οι πληροφορίες από την τρέχουσα επανάληψη του αλγορίθμου ενσωματώνονται στο πίνακα φερομόνης. Με αυτόν τον τρόπο η αποικία απεικονίζει τη γνώση που αποκτά για το περιβάλλον της. Η ευριστική πληροφορία και η προαιρετική ενσωμάτωση daemon action συνήθως με τη μορφή μιας μεθόδου τοπικής έρευνας είναι στην ευχέρεια του σχεδιαστή του αλγορίθμου και οι επιλογές που γίνονται έχουν σημαίνον ρόλο στη λειτουργία του αλγορίθμου.

Για να κατασκευαστεί μια λύση τα τεχνητά μυρμήγκια κινούνται στην αναπαράσταση του προβλήματος υπό τη μορφή ενός γραφήματος, όπου κάθε κορυφή είναι ένα πιθανό μέρος της λύσης. Ο πιθανοκρατικός κανόνας κατασκευής λύσεων χρησιμοποιείται για την επιλογή της επόμενης κορυφής που θα επισκεφθεί το μυρμήγκι. Ο αλγόριθμος σταματά όταν εξαντληθούν οι κορυφές του γραφήματος ή έχει πληρωθεί κάποιο κριτήριο τερματισμού. Πρέπει να σημειωθεί ότι δημιουργείται πλήθος λύσεων καθώς κάθε μυρμήγκι κατασκευάζει μια λύση. Αυτές ανάλογα με το αν έχει ενσωματωθεί κάποια προαιρετική daemon action εκτιμούνται και ύστερα, ο κανόνας ανανέωσης της φερομόνης προσθέτει τις πληροφορίες για τα τμήματα της καλύτερης λύσης στον πίνακα φερομόνης. Συνήθως επιλέγεται η καλύτερη λύση της τρέχουσας επανάληψης για να χρησιμοποιηθεί από τον κανόνα ανανέωση της φερομόνης .

Η οικογένεια μεταευριστικών αλγορίθμων μυρμηγκιών (Ant Colony Optimization) ανήκει στο σχετικά πρόσφατο ερευνητικό πεδίο των στοχαστικών μεταευριστικών μεθόδων. Συχνά οι αλγόριθμοι ACO έχουν εμπλουτιστεί με λειτουργίες που δεν έχουν αντίστοιχο στα πραγματικά μυρμήγκια, όπως για παράδειγμα δράσεις που επιλέγονται συνειδητά από την συλλογική συμπεριφορά των μυρμηγκιών. Είναι μια οικογένεια αλγορίθμων ιδιαίτερα επιτυχημένη καθώς έχει εφαρμοστεί σε πλήθος προβλημάτων βελτιστοποίησης και οι αρχές της στιγμεργετικής επικοινωνίας αποδεικνύουν πειραματικά την αποτελεσματικότητα τους.

Το προφανές συμπέρασμα σχετικά με την κατανόηση των λειτουργίας των αλγορίθμων ACO είναι ότι ένας αλγόριθμος της οικογένειας Ant Colony Optimization είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν. Η ρύθμιση αυτού του μηχανισμού εξαρτάται από τα διαφορετικά τμήματα που τον απαρτίζουν καθώς και τις παραμέτρους που έχει το κάθε τμήμα. Η μακροσκοπική συμπεριφορά του αλγορίθμου για παράδειγμα, εξαρτάται είτε από τον τρόπο που τα μυρμήγκια κατασκευάζουν τις λύσεις τους με βάση τον πιθανοκρατικό κανόνα μετάβασης, είτε από τη διαδικασία ανανέωσης της φερομόνης πάντα σε σχέση με το υπό έρευνα πρόβλημα. Επειδή λοιπόν αποτελούνται από σύνθετα τμήματα που αλληλεπιδρούν είναι δύσκολο να προβλέψουμε τη συμπεριφορά τους στο κάθε προβλήματα. Γενικά για μια εφαρμογή στους αλγορίθμους ACO, η δυσκολία έγκειται στο προσδιορισμό του σωστού σχεδιασμού και στη ρύθμιση του πλήθους των παραμέτρων με σκοπό τη βέλτιστη απόδοση στο υπό έρευνα πρόβλημα. Η επιλογή της σωστής περιοχής τιμών των παραμέτρων είναι ζήτημα πειραματισμού με την εφαρμογή.

Συνοπτικά, παρατηρούμε ότι οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών ACO διαθέτει αρκετά χαρακτηριστικά που όταν συνδυάζονται δημιουργούν καινοτόμες προσεγγίσεις : χρησιμοποιεί ένα *πληθυσμό* (αποικία) τεχνητών μυρμηγκιών που *κατασκευάζουν* λύσεις εκμεταλλευόμενα την *έμμεση μορφή μνήμης* που είναι οι *τεχνητές φερομόνες*.

Κεφάλαιο 5

Εναλλακτικές Μεθοδολογίες

Εισαγωγή 5.1

Η Προσομοιωμένη Ανόπτηση, οι Γενετικοί Αλγόριθμοι και η μέθοδος Tabu Search είναι παλαιότερες εναλλακτικές αλγοριθμικές μεθοδολογίες οι οποίες έχουν εφαρμοστεί σε NP hard προβλήματα βελτιστοποίησης. Οι μέθοδοι αυτοί όμως μπορούν να χρησιμοποιηθούν σε συνδυασμό με αλγορίθμους μυρμηγκιών και συνεπώς μια ανάλυση τους είναι χρήσιμη καθώς σε αρκετά προβλήματα λειτουργούν ως αλγόριθμοι τοπικής έρευνας. Δηλαδή ο ρόλος τους είναι διττός καθώς μπορούν να δρουν ανταγωνιστικά προς τους αλγορίθμους μυρμηγκιών όταν εφαρμόζονται μόνες τους, αλλά και βοηθητικός όταν δρουν ως μέθοδοι τοπικής έρευνας.

Τέλος στο κεφάλαιο αυτό παρουσιάζεται ο αλγόριθμος API , οποίος προτείνει μια διαφορετική χρήση των ιδεών που εμπνέονται από τα φυσικά μυρμήγκια και το αντίστοιχο τεχνητό μοντέλο.

5.2 Προσομοιωμένη Ανόπτηση

Η Προσομοιωμένη Ανόπτηση (Simulated Annealing) είναι μια μέθοδος που προτάθηκε από τον Kirkpatrick[1] και αναδεικνύει ομοιότητες μεταξύ της συνδυαστικής βελτιστοποίησης και της διαδικασίας της φυσικής ανόπτησης των μετάλλων. Κατά αυτή τη διαδικασία το καυτό μέταλλο ψύχεται με μεγάλη βραδύτητα και εντός του σχηματίζεται μια κρυσταλλική δομή στα μόριά του η οποία είναι και η ελάχιστη ενεργειακή κατάσταση.

[1] Kirkpatrick S, Gelatt C Jr., and Vecchi M., Optimization by simulated annealing, Science Vol 220, No 4598, May 1983, p 457-464.

Η μέθοδος προσομοιωμένης ανόπτησης αποτελεί μια επέκταση της μεθόδου που αναπτύχθηκε από τούς Metropolis[2], για να οριστεί μια κατάσταση ισορροπίας ενός συνόλου ατόμων σε μια συγκεκριμένη θερμοκρασία Τ, και βασίζεται στον τρόπο που ψύχονται και ενδυναμώνονται τα στερεά.. Ο αλγόριθμος Π.Α. δηλαδή βασίζεται στον τρόπο με τον οποίο τα στερεά κρυσταλλώνουν στο στάδιο της ισχυροποίησης /ενδυνάμωσης (annealing stage). Η διαδικασία αυτή προβλέπει ότι ένα μεταλλό, που βρίσκεται αρχικά σε υψηλή θερμοκρασία, ψύχεται με πολύ αργό ρυθμό, δίνοντας στα άτομα αρκετό χρόνο να επανατοποθετηθούν μεταξύ τους ώστε να φθάσουν σε μια χαμηλή ενεργειακή κατάσταση. Εάν η αρχική θερμοκρασία είναι πολύ χαμηλή ή η διαδικασία της ψύξης πραγματοποιείται πολύ γρήγορα, τότε μπορεί να μην επιτευχθεί η ελάχιστη δυνατή ενεργειακή κατάσταση, προκαλώντας ατέλειες και άσχημα κατασκευασμένους κρυστάλλους.

Βασικά Τμήματα

Ο αλγόριθμος έχει τέσσερα βασικά τμήματα: μια αναπαράσταση των πιθανών λύσεων, μια μέθοδο παραγωγής τυχαίων αλλαγών εντός των λύσεων, μια αντικειμενική συνάρτηση και το χρονοπρόγραμμα ανόπτησης.

Η προσομοιωμένη ανόπτηση έχει χρησιμοποιηθεί σε προβλήματα συνδυαστικής βελτιστοποίησης όπου η καταστάσεις (states) του μετάλλου αντιστοιχούν σε εφικτές λύσεις , η ενέργεια που χρησιμοποιείται σε κάθε κατάσταση αντιστοιχεί στην αντικειμενική συνάρτηση (κόστος) και τελικά η τελική ελάχιστη ενέργεια αντιστοιχεί στη καταλληλότερη λύση. Επίσης

[2] Metropolis, N. Rosenbluth, A. Rosenbluth, M. Teller, A. Teller, E. Equation of state calculations by fast computing machines. Journal of Chemical Physics, Vol 21, 1953, pp. 1087-1092.

μια αλλαγή κατάστασης του μετάλλου αντιστοιχεί σε γειτονικές λύσεις και ως παράμετρος ελέγχου χρησιμοποιείται η θερμοκρασία.

Κάθε σημείο του χώρου αναζήτησης s αντιστοιχεί σε μία κατάσταση (state) του συστήματος και η συνάρτηση $f(s)$ που πρέπει να ελαχιστοποιηθεί αντιστοιχεί σε ένα επίπεδο ενέργειας της κατάστασης. Σκοπός είναι από μια τυχαία αρχική κατάσταση το σύστημα να βρεθεί στην κατάσταση με την ελάχιστη ενέργεια.

Οι γειτονικές καταστάσεις είναι υποψήφιες ως 'νέες κινήσεις' που θα οδηγήσουν σε νέα κατάσταση με βάση το κριτήριο αποδοχής. Συνήθως οι γειτονικές καταστάσεις ορίζονται από τη φύση του υπό έρευνα προβλήματος. Στο πρόβλημα του περιοδεύοντος πωλητή ως κατάσταση του συστήματος ορίζεται μια σειρά πόλεων και ως γειτονικές λύσεις μια νέα σειρά πόλεων που κατασκευάζεται αλλάζοντας δύο διαδοχικές πόλεις ή επιλέγοντας δύο τυχαίες και αντιστρέφοντας τη σειρά.

Κριτήριο Αποδοχής

Η αποδοχή μιας νέας λύσης γίνεται με βάση το κριτήριο Metropolis το οποίο εκφράζεται με την παρακάτω εξίσωση.

$$\Pr\{Accept\ \Delta E\} = \begin{cases} e^{-\Delta E/T} & \Delta E > 0 \\ 1 & \Delta E \le 0 \end{cases}$$

Όπου: ΔΕ η διαφορά των ενεργειών των δύο καταστάσεων

T ο ρυθμιστικός παράγοντας θερμοκρασία

Από αυτή την πιθανότητα μετάβασης, είναι προφανές ότι μια κίνηση προς ανώτερη ενεργειακή κατάσταση είναι πιθανή , αλλά η πιθανότητα σχετίζεται αντίστροφα με τη άνοδο και μειώνεται καθώς η θερμοκρασία μειώνεται. Αυτό είναι και το χαρακτηριστικό που βοηθά τον αλγόριθμο να μην εγκλωβιστεί σε τοπικά βέλτιστα.

Σε νεότερες εργασίες έχει προταθεί η αντικατάσταση του κριτηρίου Metropolis για την αποδοχή νέων λύσεων. Ο Herault[3], παρατήρησε ότι σε ορισμένες μεγάλες περιπτώσεις του προβλήματος του πλανόδιου πωλητή ο αλγόριθμος Simulated Annealing απαιτεί αρκετό χρονικό διάστημα για να βρει μια καλής ποιότητας τελική λύση. Ως λύση προτείνει ένα νέο αλγόριθμο τον Rescaled Simulated Annealing. Η μόνη διαφορά αυτού του αλγόριθμου σε σχέση με τον αλγόριθμο Simulated Annealing είναι το κριτήριο αποδοχής νέων λύσεων. Με την παρακάτω εξίσωση γίνονται αποδεκτές οι νέες λύσεις.

$$P(accept\ j) = (f(j)^{1/2} - f(i)^{1/2})^2$$

με $f(j) > f(i),\ \forall i, j \in S$

[3] Herault L. (2000) Rescaled Simulated Annealing—Accelerating Convergence of Simulated Annealing by Rescaling the States Energies. Journal of Heuristics, Vol 6, Iss 2 , p. 215-252.

Όπου: $i, j \in S$ νέα λύση j, παλία λύση i που ανήκουν στο πεπερασμένο σύνολο S

$f(j), f(i)$ οι τιμές κόστους/ενέργειας της λύσης i,j

Συνεπώς στα πρώτα στάδια του αλγορίθμου όταν η θερμοκρασία είναι αρκετά υψηλή και/ή το κόστος των νέων λύσεων είναι υψηλό, ως πιο πιθανές νέες λύσεις προτείνονται οι ελάχιστες τιμές των γειτονικών λύσεων. Με τις περαιτέρω επαναλήψεις του αλγορίθμου που οδηγούν σε μείωση της θερμοκρασίας ο αλγόριθμος αρχίζει να συγκλίνει σε μια λύση , τη τελικά βέλτιστη αλλά πιο γρήγορα σε σχέση με τον κλασικό αλγόριθμο προσομοιωμένης ανόπτησης.

Πρόγραμμα Ανόπτησης

Το χρονοπρόγραμμα ψύξης (cooling schedule) ή πρόγραμμα ανόπτησης, είναι ο κρίσιμος παράγοντας για την επιτυχία του αλγορίθμου. Αρχικά τίθεται ο ρυθμιστικός παράγοντας θερμοκρασία 'Τ' σε μια υψηλή αρχική τιμή και μετά σταδιακά μειώνεται όσο αυξάνονται οι επαναλήψεις του αλγορίθμου. Σκοπός είναι αρχικά να σαρωθεί ένα ευρύ τμήμα του χώρου αναζήτησης βρίσκοντας καλές λύσεις και ύστερα σταδιακά να κατευθυνθεί προς την περιοχή με τις ελάχιστες τιμές της αντικειμενικής συνάρτησης. Όμως ένα πολύ αργό πρόγραμμα ψύξης κοστίζει σε υπολογιστικό χρόνο, ενώ ένα πιο γρήγορο μπορεί να μη φέρει καλά αποτελέσματα καθώς δεν θα έχει δοθεί επαρκής χρόνος για να εξερευνηθεί ο χώρος αναζήτησης.

Η μέθοδος Π.Α. είναι ιδιαίτερα επιτυχημένη και υπάρχουν πολλές παραλλαγές του αλγορίθμου που όλες έχουν τα τέσσερα προαναφερθέντα τμήματα. Η σχέση των Αλγορίθμων Μυρμηγκιών και της Προσομοιωμένης Ανόπτησης είναι διττή. Η μέθοδος ανταγωνίζεται τους Αλγορίθμους Μυρμηγκιών όταν εφαρμόζεται μόνη της, αλλά μπορεί να χρησιμοποιηθεί και μαζί με την Π.Α. να λειτουργεί σαν μέθοδος τοπικής έρευνας (local search). Η βασική διαφορά τους είναι ότι οι αλγόριθμοι μυρμηγκιών έχουν τον πίνακα φερομόνης που λειτουργεί ως μνήμη του αλγορίθμου ενώ η προσομοιωμένη ανόπτηση δεν κρατά κάποιο αρχείο παλιών λύσεων.

Ο van Laarhoven[4] περιγράφει τους μηχανισμούς του αλγορίθμου στο παρακάτω ψευδοκώδικα.

input : A solution s

output : A solution s and its objective value $f(s)$

$m \leftarrow 0$

repeat

    repeat

        Perturb (s,s')

        $\Delta f_{ij} = f(i) - f(j)$

        accept=false

        if $\Delta f_{ij} > 0$

            accept=true

        else

            if $e^{-\frac{\Delta f_{ij}}{T_m}} > random[0,1]$ **then**

            accept=true

[4] van Laarhoven, P. J. M. (1987). Simulated Annealing: theory and applications, Kluwer Academic Publishers, Dordrecht.

endif

endif

if accept **then**

$s \leftarrow s'$

endif

until final solution is sufficiently approached

$T_{m+1} = g(T_m)$ where g() is strategy for reducing temperature

$m = m + 1$

until stop criterion=true

return $(s, f(s))$

5.3 Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι και η γενικότερη κλάση των Εξελικτικών Αλγορίθμων έχουν τη βάση τους σε ιδέες εμπνευσμένες από τη βιολογία και οι κύριες ιδέες τους έχουν προταθεί από τον Holland[5]. Η εξελικτική θεωρεία (evolutionary theory) περιλαμβάνει σε δεσπόζουσα θέση ιδέες όπως:

[5] Holland, J.H., (1975), Adaptation in Natural and Artificial Systems, Journal of the ACM,3, 297.

- Η επιβίωση του πλέον κατάλληλου (survival of the fittest), όπου τα άτομα που δεν έχουν επαρκώς προσαρμοστεί στις υπάρχουσες συνθήκες του περιβάλλοντος τους δεν επιβιώνουν και συνεπώς τα γενετικά χαρακτηριστικά τους εξαλείφονται από το γενικό πληθυσμό.
- Περιορισμένοι Πόροι (finite resources) που τελικά αυξάνουν τον ανταγωνισμό και εγγενώς προκαλούν την προσαρμογή και την καλύτερη χρήση των πόρων.
- Αναπαραγωγή (reproduction) κατά τρόπο που να μη προσδιορίζει αυστηρά (non deterministic) τα χαρακτηρίστηκα των απογόνων.

Η τυπική δομή ενός γενετικού αλγορίθμου απαιτεί τον ορισμό:

- Μιας αναπαράστασης των υποψηφιών λύσεων
- Ένα τρόπο αξιολόγησης της καταλληλότητας(fitness) των υποψηφιών λύσεων
- Μια μέθοδο αναπαραγωγής

Η γενική μορφή ενός γενετικού αλγορίθμου σε ψευδοκώδικα δίδεται παρακάτω:

1. Επιλογή αρχικού πληθυσμού
2. Για κάθε άτομο του αρχικού πληθυσμού
3. Επανέλαβε
    1. Επιλογή των καλύτερων ατόμων από τον πληθυσμό για Αναπαραγωγή
    2. Δημιουργία νέας γενιάς με γενετικούς τελεστές (genetic operations), συνήθως με διασταύρωση (cross-over) και μετάλλαξη (mutation), και δημιούργησε Απόγονο.
    3. Εκτίμηση καταλληλότητας (fitness) για τον Απόγονο.

4. Αντικατέστησε το χειρότερο τμήμα του πληθυσμού με τον Απόγονο

4. Μέχρι <Ικανοποίησης Συνθήκης Τερματισμού >

Πολλές φορές η συνάρτηση δεν αντικατοπτρίζει αντικειμενικά την αξία των υποψήφιων λύσεων και χρειάζεται επαναπροσδιορισμό. Για παράδειγμα στο πρόβλημα του ωρολογίου προγράμματος (time scheduling) οι περισσότερες λύσεις παραβιάζουν τους περιορισμούς που θέτει το πρόβλημα. Σε αυτή τη περίπτωση η συνάρτηση δεν δύναται να εκφράζει ποιότητα αλλά εγκυρότητα πιθανών λύσεων και πρέπει να επανεκτιμηθεί.

Οι πολλές από τις αποφάσεις που πρέπει να ληφθούν για του γενετικούς αλγορίθμους είναι κοινές και για τους αλγόριθμους μυρμηγκιών. Για παράδειγμα ο ακριβής τύπος του αλγόριθμου που θα χρησιμοποιηθεί εξαρτάται από τη φύση του προβλήματος. Ερωτήματα όπως ο τρόπος αναπαράστασης του προβλήματος, το μέγεθος του πληθυσμού ανά επανάληψη και ο τρόπος συνδυασμού με μια μέθοδο τοπικής έρευνας είναι κάποια κοινά σημεία. Αλλά και οι δύο οικογένειες αλγορίθμων έχουν το θεωρητικό πρόβλημα ορισμού της σύγκλισης καθώς και του χρόνου απόκτησης της βέλτιστης λύσης, αν αυτή μπορεί να ευρεθεί.

Οι αλγόριθμοι μυρμηγκιών έχουν το πλεονέκτημα του να προσαρμόζονται καλύτερα σε προβλήματα όπου υπάρχει μια σαφής γραφική απεικόνιση του προβλήματος και ότι δεν χρειάζονται τελεστές για να δημιουργήσουν νέες λύσεις.

Αναπαράσταση Υποψήφιων Λύσεων

Η αναπαράσταση υποψήφιων λύσεων, με όποια κωδικοποίηση επιλέξουμε, αφορά ένα σύνολο πιθανών λύσεων του προβλήματος. Αυστηρή μαθηματική αναπαράσταση απαιτείτε έτσι ώστε η επεξεργασία των λύσεων να διευκολύνει τις διαδικασίες του αλγορίθμου. Συνήθως η υποψήφια λύση αναπαριστάτε σαν μια συμβολοσειρά (string) όπου το κάθε στοιχείο της ονομάζεται γονίδιο (gene). Ως μια δυαδική συμβολοσειρά (bit-string) καθορισμένου μήκους είναι η πιο απλή και διαδεδομένη μέθοδος αναπαράστασης.

Υπάρχουν διάφοροι τρόποι αναπαράστασης (κωδικοποίησης) των υποψήφιων λύσεων και συνήθως τροποποιούνται με βάση το πρόβλημα που πρέπει να λυθεί. Η καταλληλόλητα της αναπαράστασης είναι κρίσιμη για την επιτυχία του αλγορίθμου και αρκετές φορές αποδεικνύεται στην πράξη ότι ο πλέον προφανής τρόπος αναπαράστασης της λύσης δεν είναι και ο πλέον αποτελεσματικός.

Συνάρτηση Καταλληλότητας

Η συνάρτηση καταλληλόλητας προσομοιώνει το ρόλο του περιβάλλοντος και αποτελεί την πληροφορία που έχει ο αλγόριθμος για το πρόβλημα που καλείται να λύσει. Η συνάρτηση καταλληλότητας δίνει τιμές που είναι πραγματικοί αριθμοί στο διάστημα 0 ως και το 1 και έχει σκοπό να δείξει την καταλληλότητα του υπό εξέταση χρωμοσώματος. Μια τιμή ίση με 1 δίνει το θεωρητικά τέλειο χρωμόσωμα. Η τιμή της αντικειμενικής συνάρτησης είναι ο κύριος παράγοντας για την επιβίωση και αναπαραγωγή της υποψήφιας λύσης.

Επιλογή Γονέων

Η πιθανότητα να επιλεγεί ένα χρωμόσωμα για αναπαραγωγή είναι ανάλογη με την τιμή καταλληλότητας του χρωμοσώματος. Ένας διαδεδομένος τρόπος επιλογής είναι η μέθοδος της ρουλέτας. Με αυτή τη μέθοδο το σύνολο των τιμών καταλληλότητας του πληθυσμού απεικονίζεται ως μια ρόδα ρουλέτας όπου το τόξο που καταλαμβάνει κάθε ένα χρωμόσωμα είναι ανάλογο της τιμής καταλληλότητάς του. Με τυχαία περιστροφή της ρόδας επιλέγουμε το χρωμόσωμα. Συνεπώς, τα χρωμοσώματα (δυαδική συμβολοσειρά) με τη μεγαλύτερη απόδοση θα έχουν περισσότερους απογόνους στην επόμενη γενιά ,ενώ αυτά με χαμηλή απόδοση εξαλείφονται.

Αναπαραγωγή

Το σύνολο των επιλεγμένων χρωμοσωμάτων (δυαδικές συμβολοσειρές) δημιουργεί ένα νέο δοκιμαστικό πληθυσμό που καλείται από τον Goldberg[6] δεξαμενή ζευγαρώματος (mating pool). Σε αυτόν τον πληθυσμό θα εφαρμοστούν οι γενετικοί τελεστές.

Οι γενετικοί τελεστές που χρησιμοποιούνται στις περισσότερες εφαρμογές γενετικών αλγορίθμων είναι η διασταύρωση (cross-over) και η μετάλλαξη (mutation).

Διασταύρωση

Με τη μέθοδο της διασταύρωσης (cross-over) για κάθε ζεύγος επιλεγμένων χρωμοσωμάτων γονέων διαλέγουμε τυχαία ένα σημείο διασταύρωσης (cross-point) που ορίζετε ένας αριθμός N από το 1 μέχρι το L , όπου L το μήκος της συμβολοσειράς. Με γονείς τους Γ1 και Γ2, ο ένας απόγονος Α1 δημιουργείτε από τα πρώτα N γονίδια (bits) του Γ1 και τα L-N του Γ2. Αντίστοιχα ο απόγονος Α2 δημιουργείτε από τα πρώτα N γονίδια (bytes) του Γ2 και τα L-N του Γ1.

Μετάλλαξη

[6] Goldberg D.E.,GENETIC ALGORITHMS in Search, Optimization and Machine Learning, Addison Wesley Publishing Company, Inc., 1989

Ο τελεστής μετάλλαξη (mutation) εκτελείτε στους απογόνους της διασταύρωσης. Καθώς αντιγράφονται τα γονίδια (bits) για την παραγωγή νέων χρωμοσωμάτων (strings), με μια μικρή τυχαία πιθανότητα αντιστρέφεται ένα ψηφίο από 0 σε 1 ή το αντίστροφο. Αυτή η πιθανότητα καλείται πιθανότητα μετάλλαξης (mutation probability).

Στην πρακτική εφαρμογή της διασταύρωσης χρησιμοποιούμαι τη πιθανότητα διασταύρωσης (p_c) που ορίζεται από τον ερευνητή. Η τιμή της πιθανότητας διασταύρωσης επηρεάζει το χρόνο τρεξίματος του αλγορίθμου καθώς όταν p_c =1, ο τελεστής εφαρμόζεται συνεχώς σε κάθε επανάληψη. Αυτό σημαίνει ότι ερευνάμε όλο το χώρο αναζήτησης και θα συγκλίνουμε στο βέλτιστο αλλά με πολύ αργά. Πολλές φορές στην αρχή της εκτέλεσης επιλέγεται μικρή τιμή της πιθανότητας διασταύρωσης και καθώς συγκλίνουμε προς τη βέλτιστη λύση αυξάνουμε την τιμή της ώστε να έχουμε και βέλτιστο αποτέλεσμα και μικρό χρόνο εκτέλεσης.

Η μετάλλαξη εξασφαλίζει ότι κανένας τομέας του χώρου αναζήτησης δεν αποκλείεται από τη διαδικασία έρευνας. Επίσης προστατεύει από την πιθανότητα κατά την επιλογή και τη διασταύρωση να απολεσθούν χρήσιμα γονίδια μέσα στα χρωμοσώματα. Έχει παρατηρηθεί ότι αν η πιθανότητα μετάλλαξης είναι υψηλή, ο αλγόριθμος εκφυλίζεται σε τυχαία έρευνα. Συνεπώς η πιθανότητα μετάλλαξης πρέπει να είναι ιδιαίτερα μικρή και σαν καλή πρακτική προτείνεται συνήθως μια μετάλλαξη αν χίλια ψηφία που αντιγράφονται.

Οι Γενετικοί Αλγόριθμοι και οι Αλγόριθμοι Μυρμηγκιών έχουν ένα σύνολο επιλογών στο σχεδιασμό των αλγορίθμων που είναι κοινό, όπως για παράδειγμα η αναπαράσταση του προβλήματος, το μέγεθος του πληθυσμού των μυρμηγκιών ανά επανάληψη και η χρήση ή μη μεθόδου τοπική έρευνας. Το μεγαλύτερο βάρος όμως στο σχεδιασμό του αλγορίθμου τον θέτει το υπό έρευνα πρόβλημα και σε αυτό, οι αλγόριθμοι μυρμηγκιών έχουν το πλεονέκτημα να ταιριάζουν καλύτερα σε προβλήματα που η περιγραφή τους στηρίζεται σε γράφημα. Επίσης η κατασκευαστική φύση των αλγορίθμων μυρμηγκιών δεν χρειάζεται τελεστές μετάβασης για να δημιουργήσει νέες λύσεις.

5.4 Tabu Search

Είναι μια μέθοδος αναζήτησης η οποία, όπως και με την μέθοδο της Προσομοιωμένης Ανόπτησης, μπορεί να χρησιμοποιηθεί είτε ως μέθοδος τοπικής έρευνας είτε ως μεταευριστική μέθοδος και έχει προταθεί από τον Glover[7].

Παραδοσιακά η αντίληψη ότι κοντά στις καλές λύσεις υπάρχουν οι ακόμα καλύτερες έχει δημιουργήσει αλγορίθμους που επικεντρώνονται στην αναζήτηση σε μια σχετικά περιορισμένη γειτονιά λύσεων. Για αυτό το λόγο η βασική ιδέα στην Tabu Search είναι η δημιουργία μιας μνήμης των καλών λύσεων, ώστε πιθανή αναζήτηση νέων καλύτερων λύσεων να μην τιμωρεί τον αλγόριθμο ξεχνώντας την παλαιότερη καλή λύση.

Ο αλγόριθμος ξεκινά δημιουργώντας μια τυχαία λύση $x_0 \in X$ η οποία βελτιώνεται μετά από επαναλήψεις. Κατά την εκτέλεση του αλγορίθμου υπάρχει, όπως και στον αλγόριθμο Προσομοιωμένης Ανόπτησης, μια τρέχουσα λύση (current solution) και μια καλύτερη έως τώρα (best so far). Έστω ότι στη επανάληψη i δημιουργήθηκε η τρέχουσα λύση x_i, τότε για να αποκτήσουμε μια νέα λύση που ανήκει στο X πρέπει να γίνει κίνηση (move) τροποποίησης της

[7] Glover F. (1986) Future paths for integer programming and links to artificial intelligence, Computers and Operation Research, 13, 533-549.

x_j. Αυτή είναι μια γειτονική της λύσης x_i και το σύνολο των γειτονικών λύσεων ορίζεται ως $N(x_i)$, με $N(x_i) \subset X$ και $x_i \notin N(x_i)$.

Είναι πιθανό η λύση που βρέθηκε να είναι το τοπικό ελάχιστο. Συνεπώς το να μεταβεί ο αλγόριθμος σε κάποια άλλη λύση, που με βάση την αντικειμενική συνάρτηση δεν θα είναι καλύτερη, είναι φαινομενικά λάθος συμπεριφορά του αλγορίθμου. Όμως αυτή η 'λάθος' συμπεριφορά του αλγορίθμου επιτρέπεται, καθώς στους αλγόριθμους Tabu Search δύναται το σύστημα να αποδεχθεί μια τρέχουσα λύση με χειρότερη τιμή της ολικά καλύτερης, έτσι ώστε να γίνεται εξερεύνηση της γειτονιάς της λύσης $N(x_i)$. Αν δηλαδή 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.

Έτσι λοιπόν, γίνεται χρήση μνήμης στον αλγόριθμο όπου οι καλές λύσεις σημειώνονται σε μια λίστα Tabu , από όπου και το όνομα της μεθόδου, τις οποίες κατά τη φάση κινήσεων για νέες λύσεις δεν πρέπει να επισκεφθεί ο αλγόριθμος. Δηλαδή για ένα αριθμό επαναλήψεων i δεν μπορούμε να επιστρέψουμε στις λύσεις της Tabu list, και συνεπώς μια γειτονιά λύσεων είναι σαφέστερα η $N(x_i, i)$. Ανάλογα με το σχεδιασμό του αλγορίθμου, η χρήση μνήμης περιέχει μια history list και tabu list οι οποίες τιμωρούν ή απαγορεύουν κινήσεις και κατευθύνουν την αναζήτηση σε πρόσφατες λύσεις.

Παρακάτω δίδεται ο ψευδοκώδικας[8] της μεθόδου Tabu Search ,όπου x^* η βέλτιστη λύση.

[8] Dammeyer F, Voß S (1993). Dynamic tabu list management using the reverse elimination method. Annals of Operation Research, 41, 31-46.

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 $F^* := F(x_{i+1})$;

Increment (i);

End While

Όταν η Tabu Search χρησιμοποιείται ως μέθοδος τοπικής έρευνας σε συνδυασμό με αλγορίθμους ACO αποδίδει εξαιρετικά αποτελέσματα όπως έδειξε ο Stutzle[9] σε δύσκολα προβλήματα τετραγωνικής ανάθεσης. Χρήση της μεθόδου σε στοχευμένες περιοχές του χώρου αναζήτησης δίνει στους αλγορίθμους μυρμηγκιών την ευκαιρία να εμφανίσουν το μεταευριστικό τους πλεονέκτημα καθοδηγώντας στρατηγικά την έρευνα και για αυτό το λόγο πολλοί ερευνητές μελετούν τη συνδυασμένη χρήση τους. Αλλά ένα μειονέκτημα της Tabu Search είναι η επιλογή

[9] Stutzle, T. (1997). MAX-MIN Ant System for Quadratic Assignment Problems. Technical Report AIDA.97.4, FG Intellektik, TU Darmstadt, Germany.

του σωστού σχεδιασμού και των σωστών παραμέτρων, πράγμα που δυσκολεύει ακόμα περισσότερο το συνδυασμό της με αλγορίθμους μυρμηγκιών.

5.5 Αλγόριθμος API

Ο αλγόριθμος API προτάθηκε από τον Monmarche[10] και οφείλει το όνομα του στο είδος μυρμηγκιών Pachychondyla Apicalis. Πρόκειται για ένα αλγόριθμο που εξηγεί ένα νέο τρόπο με τον οποίο αναζητούν τροφή τα μυρμήγκια καθώς διαφέρει από τον τρόπο αναζήτησης που έχουμε προαναφέρει. Η στρατηγική τους είναι απλή αλλά αποτελεσματική καθώς κάθε ένα μυρμήγκι ξεχωριστά εξερευνά τη περιοχή γύρω από τη φωλιά. Δηλαδή εκτελούν μαζική παράλληλη αναζήτηση σε όλο το χώρο αναζήτησης αλλά με ευαισθησία σε επιτυχημένες περιοχές αναζήτησης. Σημαντικό είναι και το γεγονός ότι η φωλιά της αποικίας αλλάζει θέση κατά την αναζήτηση.

Σε κάθε αποικία μόνο το 20% με 30% του πληθυσμού βρίσκεται εκτός της φωλιάς για αναζήτηση τροφής. Σημείο διαφοροποίησης με το σύστημα ACO είναι ότι το κάθε μυρμήγκι δεν επικοινωνεί με φερομόνη με τα υπόλοιπα και ότι έχει ξεχωριστή μνήμη στην οποία καταγράφει τις επιτυχημένες περιοχές που βρέθηκε τροφή. Η συνεργασία της αποικίας εκτελείται με μία διαδικασία στρατολόγησης. Ο συνολικός χώρος αναζήτησης είναι συνήθως ένας κύκλος με ακτίνα 10 μέτρα γύρω από τη φωλιά. Η περιοχή που ερευνά τοπικά κάθε μυρμήγκι ξεχωριστά είναι ένας κύκλος με ακτίνα περίπου 2,5 μέτρα. Αν η τοπική αναζήτηση φέρει ένα επιτυχημένο αποτέλεσμα, δηλαδή ανακαλύψει τροφή, κρατά στη μνήμη του την περιοχή. Διαφορετικά

[10] Monmarché N., Venturini G., Slimane M., On how Pachycondyla apicalis ants suggest a new search algorithm, Future Generation Computer Systems 16 (2000) p 937–946.

προχωρά στην εξερεύνηση νέας περιοχής. Σε περίπτωση που δεν υπάρχει αρκετή τροφή στην ευρύτερη περιοχή της φωλιάς, η φωλιά μετακινείται.

Ο ψευδοκώδικας παρακάτω δείχνει τους μηχανισμούς του αλγορίθμου.

Process API

Choose randomly the initial location of the nest N

While stopping criterion not reached **do**

For each ant $a_i, i \in [1,...n]$

if a_i has less than p hunting sites in memory **then**

Create new site in the neighborhood of N

Explore new site

else

if the previous site exploration was successful **then**

Explore again the same site

else

Select randomly and uniformly another site among p sites in memory

Explore a randomly selected site

```
            endif
      else
            Remove from ants memories all sites which have been explored unsuccessfully
            for more than P_local(a_i) consecutive times
      endif
      Perform recruitment (best site copying between two randomly selected ants)
      if more than T iterations have been performed then
            Change nest location
            Reset the memories of all ants
      endif
endw
```

Η περιοχή αναζήτησης των μυρμηγκιών αντιστοιχεί στο χώρο αναζήτησης του προβλήματος.

Θεωρούμε ένα πλήθος n μυρμηγκιών $a_1,\dots,a_n$ τα οποία βρίσκονται στο χώρο αναζήτησης S και έχουν ως σκοπό την ελαχιστοποίηση της αντικειμενικής συνάρτησης $f: S \to \mathbb{R}$. Κάθε σημείο $s \in S$ είναι λύση του προβλήματος και το S μπορεί να είναι συνεχής χώρος ($S = \mathbb{R}^l$), δυαδικός χώρος ($S = \{0,1\}^l$) ή χώρος μεταθέσεων όπως στο πρόβλημα του περιοδεύοντος πωλητή. Στο πλαίσιο API ο χώρος αναζήτησης είναι σχετικός με το πρόβλημα.

Η εφαρμογή του αλγορίθμου ορίζει τους δύο παρακάτω τελεστές:

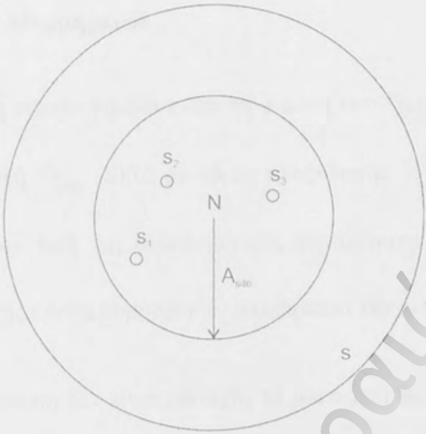
O_{rand} : δημιουργεί ένα τυχαίο σημείο στο S σύμφωνα με μια ομοιόμορφη κατανομή

O_{explo} : δημιουργεί ένα σημείο s' στη γειτονία του s

Το μέγεθος της γειτονιάς του γύρω από το s τίθεται ίσο με τιμή A , $A \in [0,1]$. Η τιμή αυτή ορίζει το εύρος της αναζήτησης. Σημαντικό είναι ότι ο O_{explo} μπορεί να μην είναι τυχαίος αλλά να βασίζεται σε κάποια ευριστική μέθοδο που είναι σχετική με το χώρο αναζήτησης.

Εξερεύνηση του Χώρου Αναζήτησης

Με την έναρξη του αλγόριθμος επιλέγει τυχαία την τοποθεσία της φωλιάς N με βάση τον O_{rand}. Σε κάθε επανάληψη τα μυρμήγκια κινούνται παράλληλα. Σε κάθε T επαναλήψεις η θέση της φωλιάς αλλάζει και επιλέγεται νέα τοποθεσία που είναι η καλύτερη που έχει ανακαλυφθεί μέχρι εκείνη τη στιγμή.



Σχήμα 1. Τα σημεία s1 ,s2 και s3 επιλέγονται τυχαία και η μέγιστη απόσταση που μπορούν να έχουν από τη φωλιά N είναι η Asite.

Κάθε μυρμήγκι a_i ξεκινά από τη φωλιά για να δημιουργήσει τυχαία p περιοχές αναζήτησης στη γειτονιά του N. Αυτό γίνεται με τη χρήση του τελεστή O_{explo} με εύρος αναζήτησης A που ορίζεται ως $A_{site}(a_i)$.

$A_{site}(1) = 0.01, ...,$

$A_{site}(i) = x^i \cdot 0.01, ...$

$A_{site}(n) = x^n \cdot 0.01 = 1$

όπου $x = (1/0,01)^{1/n}$

Ο αριθμός των θέσεων που μπορεί έχει στη μνήμη ένα μυρμήγκι είναι p= 2.

Τοπική Συμπεριφορά των Μυρμηγκιών

Το μυρμήγκι a_i επιλέγει το τυχαίο σημείο s από τα p μέρη που έχει στη μνήμη Σε αυτό ερευνά τοπικά το σημείο με τελεστή O_{explo} αλλά με εύρος αναζήτησης $A_{local}(a_i)$. Η αναζήτηση αυτή δίνει το σημείο s' και την τιμή της αντικειμενικής συνάρτησης $f(s')$. Εάν $f(s') < f(s)$ η εξερεύνηση κρίνεται επιτυχής και το μυρμήγκι a_i αποθηκεύει την τοποθεσία $s \leftarrow s'$.

Σε περίπτωση που η εξερεύνηση δεν είναι επιτυχής το μυρμήγκι θα αναζητήσει ένα νέο τυχαίο σημείο από τα p που έχει στη μνήμη του. Εάν μετά από $P_{local}(a_i)$ αναζητήσεις το σημείο δεν έχει δώσει επιτυχημένα αποτελέσματα, το σημείο αφαιρείται από τη μνήμη του μυρμηγκιού και η θέση του λαμβάνει νέο που δημιουργείται από τον O_{explo} με εύρος αναζήτησης $A_{local}(a_i)$. Επίσης μετά από κάθε μετακίνηση της φωλιάς, σβήνονται οι μνήμες όλων των μυρμηγκιών. Αυτό το χαρακτηριστικό έχει σκοπό την αποφυγή εγκλωβισμού του αλγορίθμου σε τοπικό βέλτιστο.

Σχήμα 2. Τα τρίγωνα στο σχήμα αναπαριστούν τις τοπικές εξερευνήσεις γύρω από το σημείο s1 με μέγιστη απόσταση από αυτό Alocal.

Προτείνεται ότι : $A_{local}(a_i) = A_{suc}(a_i)/10$.

Συνεργασία των Μυρμηγκιών

Η άμεση συνεργασία των μυρμηγκιών εκφράζεται με την μετακίνηση της φωλιάς, καθώς από την ατομική αναζήτηση ενός μυρμηγκιού προκύπτουν νέες καλύτερες τοποθεσίες για τη φωλιά. Η ανταλλαγή αυτής της πληροφορίας οδηγεί στη μετακίνηση της φωλιάς. Επίσης η στρατολόγηση μεταξύ των μυρμηγκιών έχει πολύ σημαντικό ρόλο. Κάθε φορά που η μυρμήγκια εξερευνούν , επιλέγονται δύο μυρμήγκια a_i και a_j τυχαία. Έστω ότι η καλύτερη περιοχή του a_i δίνει αποτέλεσμα καλύτερο της καλύτερης περιοχής του a_j . Τότε η νέα καλύτερη περιοχή του a_j θα γίνει η ίδια με την καλύτερη περιοχή του a_i . Έτσι γίνεται υψηλότερη εκμετάλλευση μιας περιοχής καθώς αυξάνεται ο αριθμός των αναζητήσεων που επιτρέπεται στη συγκεκριμένη περιοχή, αλλά και γιατί διαφορετικά μυρμήγκια μπορεί να έχουν άλλες παραμέτρους τοπικής αναζήτησης.

5.6 Συμπεράσματα

Οι εναλλακτικές μεθοδολογίες που παρουσιάζονται στη παρούσα εργασία είναι αλγόριθμοι οι οποίοι μπορούν να εφαρμοστούν είτε μόνες τους είτε ως μέθοδοι τοπικής έρευνας και στη διεθνή βιβλιογραφία υπάρχουν δημοσιεύσεις όπου οι μέθοδοι υλοποιούνται είτε με τον ένα είτε με τον άλλο τρόπο, αλλά όχι πάντα με επιτυχημένα αποτελέσματα. Στο συνδυασμό για παράδειγμα της Tabu Search με κάποιο αλγόριθμο μυρμηγκιών η επιλογή του σωστού σχεδιασμού και των σωστών παραμέτρων είναι κρίσιμη, πράγμα που δυσκολεύει ακόμα περισσότερο το συνδυασμό της με αλγορίθμους μυρμηγκιών.

Ένα άλλο παράδειγμα είναι η μέθοδος Προσομοιωμένης Ανόπτησης που είναι ιδιαίτερα επιτυχημένη και υπάρχουν πολλές παραλλαγές του αλγορίθμου. Η σχέση των Αλγορίθμων Μυρμηγκιών και της Προσομοιωμένης Ανόπτησης είναι διττή. Η μέθοδος ανταγωνίζεται τους Αλγορίθμους Μυρμηγκιών όταν εφαρμόζεται μόνη της, αλλά μπορεί να χρησιμοποιηθεί και μαζί με την Π.Α. να λειτουργεί σαν μέθοδος τοπικής έρευνας (local search). Η βασική διαφορά τους είναι ότι οι αλγόριθμοι μυρμηγκιών έχουν τον πίνακα φερομόνης που λειτουργεί ως μνήμη του αλγορίθμου ενώ η προσομοιωμένη ανόπτηση δεν κρατά κάποιο αρχείο παλιών λύσεων.

Η φύση του υπό έρευνα προβλήματος και ο σχεδιασμός του κάθε αλγορίθμου σχεδόν προκαθορίζουν την απόδοση μιας αλγοριθμικής εφαρμογής. Οι Γενετικοί Αλγόριθμοι και οι Αλγόριθμοι Μυρμηγκιών έχουν ένα σύνολο επιλογών στο σχεδιασμό των αλγορίθμων που είναι κοινό, όπως για παράδειγμα η αναπαράσταση του προβλήματος, το μέγεθος του πληθυσμού των μυρμηγκιών ανά επανάληψη και η χρήση ή μη μεθόδου τοπική έρευνας. Το μεγαλύτερο βάρος όμως στο σχεδιασμό του αλγορίθμου τον θέτει το υπό έρευνα πρόβλημα και σε αυτό, οι αλγόριθμοι μυρμηγκιών έχουν το πλεονέκτημα να ταιριάζουν καλύτερα σε προβλήματα που η περιγραφή τους στηρίζεται σε γράφημα. Επίσης η κατασκευαστική φύση των αλγορίθμων μυρμηγκιών δεν χρειάζεται τελεστές μετάβασης για να δημιουργήσει νέες λύσεις.

Τέλος, ο αλγόριθμος API εξαρτάται σε μεγάλο βαθμό από τον τελεστή $O_{exp/o}$ και είναι πολύ διαφορετικός από τους αλγόριθμους ACO. Είναι όμως ιδιαίτερα εύκολος να προσαρμοστεί σε διαφορετικούς χώρους αναζήτησης σε σχέση με τους αλγόριθμους ACO αλλά δεν έχει μεγάλο βαθμό αποτελεσματικότητας.

Κεφάλαιο 6

Βελτιστοποίηση Θαλάσσιων Μεταφορών

με τον Αλγόριθμο Προσομοιωμένης Ανόπτησης

Εισαγωγή 6.1

Η Ελληνική επικράτεια έχει 14.854 χιλιόμετρα ακτογραμμής, περίπου 3.500 νησιά (εκ των οποίων 200 είναι ακατοίκητα) και 750 λιμένες. Αυτά τα νησιά αποτελούν το 19% της έκτασης της χώρας και κατοικούνται από το 14% του πληθυσμού. Η μεταφορά αγαθών σε αυτά περατώνεται με φορτηγά αυτοκίνητα τα οποία φθάνουν στον προορισμό τους με επιβατηγά/οχηματαγωγά πλοία και μικρά πλοία εμπορευμάτων.

Θεωρούμε μία ομάδα δεκατριών (13) λιμένων που είναι υποσύνολο των λιμένων του Αιγαίου πελάγους, η οποία απεικονίζεται στο γράφημα 1. Τα τόξα που συνδέουν τους κόμβους του γραφήματος αντιστοιχούν σε υπαρκτές ή εφικτές θαλάσσιες διαδρομές ανάμεσα σε αυτά τα νησιά. Σκοπός είναι με συγκεκριμένες συνθήκες να ελαχιστοποιηθεί το κόστος των καυσίμων για ένα πλοίο που επισκέπτεται μία φορά ένα λιμένα και το κόστος χρήσης του κάθε λιμένα. Σαν μέθοδοι επίτευξης αυτού του σκοπού χρησιμοποιούνται ο αλγόριθμος Προσομοιωμένης Ανόπτησης (Simulated Annealing Algorithm) και ο αλγόριθμος του Floyd (Floyds Algorithm).

Το γράφημα που προκύπτει από τον πίνακα με το κόστος μετάβασης πρέπει να είναι ισχυρά συνεκτικό . Συνεπώς από κάθε κόμβο/λιμάνι πρέπει να είναι δυνατή η μετάβαση σε οποιοδήποτε άλλο κόμβο/λιμάνι με σαφές κόστος μετάβασης και οπότε όλα τα στοιχεία του πίνακα μετάβασης πρέπει να έχουν τιμές μεγαλύτερες του μηδενός εκτός της κυρίου διαγωνίου του πίνακα που πρέπει να είναι μηδέν.

Η λύση του προβλήματος γίνεται περισσότερο κατανοητή καθώς προσεγγίζεται με τη χρήση δύο μεθόδων. Χρησιμοποιούμε τον αλγόριθμο του Floyd για τον προσδιορισμό των ελάχιστων αποστάσεων μεταξύ των κόμβων του γραφήματος είτε με άμεση μετάβαση σε ένα κόμβο, είτε με μετάβαση μέσω ενός άλλου κόμβου. Σημαντικό είναι ότι ο αλγόριθμος του Floyd βρίσκει το ελάχιστο κόστος μετάβασης για κάθε κόμβο/ λιμάνι ξεχωριστά και δεν προσδιορίζει το ολικό κόστος μετάβασης που βρίσκεται με τη βοήθεια του αλγόριθμου Προσομοιωμένης Ανόπτησης.

6.2 Έκφραση Προβλήματος

Επιλέγουμε ένα υποσύνολο των νησιών του Αιγαίου που απεικονίζεται στο γράφημα 1. Το γράφημα αναπαριστά τις αποστάσεις ανάμεσα σε αυτά τα 13 νησιά όπως δίδονται από το Υπουργείο Εμπορικής Ναυτιλίας. Τα τόξα που συνδέουν τους κόμβους είναι οι εφικτές εμπορικές γραμμές ανάμεσα στα νησιά. Για κάθε μία από τις εφικτές εμπορικές γραμμές υπάρχει ένα κόστος μετάβασης που είναι ανάλογο της απόστασης των νησιών που ενώνει. Για κάθε ναυτικό μίλι αυτής της απόστασης το πλοίο καταναλώνει συγκεκριμένη ποσότητα καυσίμου, ενώ για το ολικό κόστος μετάβασης πρέπει να προσθέσουμε και τα έξοδα χρήσης του λιμένα.

Υποθέτουμε ότι το κόστος χρήσης του λιμένα του Πειραιά είναι μηδέν και είναι και ο λιμένας αφετηρία των διαδρομών. Επίσης η απόσταση ανάμεσα σε δύο κόμβους του γραφήματος είναι ίδια και στις δύο κατευθύνσεις, άρα το κόστος μετάβασης από το λιμένα i στο λιμένα j είναι το ίδιο με το κόστος μετάβασης από το λιμένα j στο λιμένα i.

Η αντικειμενική συνάρτηση του προβλήματος δίδεται παρακάτω.

$$\min \sum_{\text{all arcs ij}} \left[L_{ij} \cdot c_f + c_{pi} \right] \tag{4.1}$$

Όπου:

L_{ij} : το μήκος του τόξου i,j

c_f : το κόστος των καυσίμων που καταναλώνονται ανά ναυτικό μίλι

c_{pi} : το κόστος χρήσης του λιμένα i

Επίσης θεωρούμε ότι:

$L_{ij} \geq 0$, ∀ arc (i,j): περιορισμός μη αρνητικότητας (4.2)

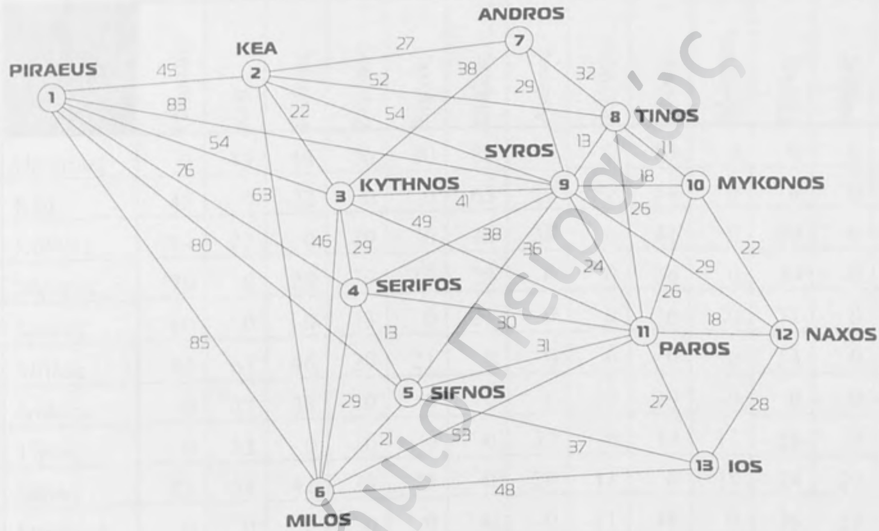
$c_f \geq 0$, ∀ arc (i,j): περιορισμός μη αρνητικότητας (4.3)

$c_{pi} \geq 0$, ∀ arc (i,j): περιορισμός μη αρνητικότητας (4.4)

$c_{p1} = 0$, παραδοχή ότι το κόστος χρήσης του λιμένα του Πειραιά (λιμένας αφετηρίας) είναι μηδέν (4.5)

$L_{ij} = L_{ji}$, ∀ arc (i,j): παραδοχή ότι το τόξο (i,j) έχει ίσο μήκος με το τόξο (j,i) (4.6)

Για τα 13 αυτά νησιά το L_{ij} λαμβάνεται από τον πίνακα με το κόστος μετάβασης του Υ.Ε.Ν. και προκύπτει το παρακάτω γράφημα.



Γράφημα 1: Το επιλεγμένο σύνολο των 13 κόμβων/λιμένων και οι αποστάσεις τους.

Ο πίνακας γειτνίασης (adjacency matrix) που προκύπτει δίδεται παρακάτω.

		1	2	3	4	5	6	7	8	9	10	11	12	13
		Πειραιάς	Κέα	Κύθνος	Σέριφος	Σίφνος	Μήλος	Άνδρος	Τήνος	Σύρος	Μύκονος	Πάρος	Νάξος	Ίος
1	Πειραιάς	0	45	54	76	80	85	0	0	83	0	0	0	0
2	Κέα	45	0	22	0	0	63	27	52	54	0	0	0	0
3	Κύθνος	54	22	0	29	0	46	38	0	41	0	49	0	0
4	Σέριφος	76	0	29	0	13	29	0	0	38	0	30	0	0
5	Σίφνος	80	0	0	13	0	21	0	0	36	0	31	0	37
6	Μήλος	85	63	46	29	21	0	0	0	0	0	53	0	48
7	Άνδρος	0	27	38	0	0	0	0	32	29	0	0	0	0
8	Τήνος	0	52	0	0	0	0	32	0	13	11	26	0	0
9	Σύρος	83	54	41	38	36	0	29	13	0	18	24	29	0
10	Μύκονος	0	0	0	0	0	0	0	11	18	0	26	22	0
11	Πάρος	0	71	49	30	31	53	0	26	24	26	0	18	27
12	Νάξος	0	0	0	0	0	0	0	0	29	22	18	0	28
13	Ίος	0	0	0	0	37	48	0	0	0	0	27	28	0

Πίνακας 1: Ο πίνακας γειτνίασης Α

	Λιμάνι	Κόστος Χρήσης c_{pi}
1	Πειραιάς	0
2	Κέα	28
3	Κύθνος	30
4	Σέριφος	25
5	Σίφνος	39
6	Μήλος	19
7	Άνδρος	37
8	Τήνος	20
9	Σύρος	35
10	Μύκονος	22
11	Πάρος	21
12	Νάξος	40
13	Ίος	30

Πίνακας 2 : Ο πίνακας με το κόστος χρήσης των λιμένων

Το κόστος των καυσίμων τίθεται ίσο με 5 για κάθε μίλι. $c_f = 5$

6.3 Ο Αλγόριθμος Προσομοιωμένης Ανόπτησης

Ο αλγόριθμος Προσομοιωμένης Ανόπτησης προτάθηκε από τον Kirkpatrick[1] και η προέλευση του βρίσκεται στη στατιστική μηχανική (statistical mechanics) και βασίζεται στο φυσικό φαινόμενο της επανισορρόπησης των μορίων ενός μετάλλου σε φυσικοχημικό επίπεδο ενώ αυτό ψύχεται αργά. Σε υψηλές θερμοκρασίες τα μόρια του μετάλλου κινούνται σχετικά ελεύθερα. Εάν

[1] Kirkpatrick S, Gelatt C Jr., and Vecchi M., Optimization by simulated annealing, Science Vol 220, No 4598, May 1983, p 457-464.

επιτρέψουμε στα μόρια να κινούνται με σταδιακά μειούμενη κινητική ενέργεια, το σύστημα τείνει σε μια κατάσταση όπου σχηματίζονται τέλειοι κρύσταλλοι και η ελάχιστη κατάσταση ενέργειας (lowest energy state) έχει επιτευχτεί. Οι αντιστοιχίες μεταξύ των καταστάσεων ενέργειας (energy states) και της αντικειμενικής συνάρτησης (objective function) σε προβλήματα βελτιστοποίησης είναι προφανής. Η Dowsland[2] υποστηρίζει ότι, η αντιστοιχία μπορεί να χρησιμοποιηθεί σε προβλήματα συνδυαστικής βελτιστοποίησης όπου οι καταστάσεις του στερεού αντιπροσωπεύουν την εφικτή λύση, η ενέργεια σε κάθε κατάσταση αντιπροσωπεύει τη βελτίωση της αντικειμενικής συνάρτησης και η ελάχιστη ενέργεια την καταλληλότερη λύση. Ο παρακάτω πίνακας συνοψίζει τις αντιστοιχίες.

Θερμοδυναμική Προσομοίωση	**Συνδυαστική Βελτιστοποίηση**
Καταστάσεις Συστήματος	Εφικτές Λύσεις
Ενέργεια	Κόστος
Αλλαγή Κατάστασης	Γειτονικές Λύσεις
Θερμοκρασία	Παράμετρος Ελέγχου
Κατάσταση Ψύξης	Βέλτιστη Λύση

Πίνακας 3 : Αντιστοιχίες θερμοδυναμικής προσομοίωσης και συνδυαστικής βελτιστοποίησης

Από μια δεδομένη κατάσταση, στην οποία τα μόρια είναι διαταγμένα σε συγκεκριμένες θέσεις, δημιουργούμε μία μικρή διαταραχή με σκοπό να επαναδιαταχθούν πιθανοκρατικά σε μία νέα κατάσταση η οποία πιθανώς χρειάζεται χαμηλότερη ενέργεια. Αυτή η διαφορά ΔΕ στην ενεργειακή κατάσταση, εάν είναι αρνητική οδηγεί σε αποδοχή της νέας κατάστασης και η

[2] Dowsland K.A., Off-the-Peg or Made-to-Measure? Timetabling and Scheduling with SA and TS, in Practice and Theory of Automated Timetabling II, Second International Conference, PATAT'97, Toronto, Canada, August 20-22, 1997, Selected Papers. Lecture Notes in Computer Science 1408 Springer 1998 p: 37-52.

διαδικασία συνεχίζει. Εάν η ΔΕ είναι θετική, τότε η νέα κατάσταση γίνεται αποδεκτή με πιθανότητα ίση με $\exp\{-\Delta E / t\}$. Αυτό είναι και το κριτήριο αποδοχής Metropolis[3].

$$\Pr\{Accept\,\Delta E\} = \begin{cases} e^{-\Delta E/t} & \Delta E > 0 \\ 1 & \Delta E \leq 0 \end{cases} \tag{4.7}$$

Επίσης, η σταθερή κατάσταση των πιθανοτήτων $\pi_i(t)$ για μία κατάσταση i δίδεται από τον τύπο:

$$\pi_i(t) = \frac{e^{-E_i/t}}{B(t)} \tag{4.8}$$

Όπου: B(t) η συνάρτηση Boltzmann

E_i το επίπεδο ενέργειας στην κατάσταση i

Για να συμβεί η μετάβαση από τη μια κατάσταση στην άλλη, το κριτήριο Metropolis επαναδιατυπώνεται ως :

$$p_{ij}(t_k) = \begin{cases} G_{ji} e^{-\Delta f^+_{ji}/t_k} & \Delta f^+_{ji} > 0 \\ 1 & \Delta f^+_{ji} \leq 0 \end{cases} \tag{4.9}$$

με G_{ji} την πιθανότητα να δημιουργηθεί η υποψήφια κατάσταση j με δοσμένη την κατάσταση i, που σημαίνει ότι η j είναι στη γειτονία του i , $j \in N(i)$, και Δf^+_{ji}, $\Delta f^+_{ji} \equiv \max\{0, f_j - f_i\}$.

[3] Metropolis, N. Rosenbluth, A. Rosenbluth, M. Teller, A. Teller, E. Equation of state calculations by fast computing machines. Journal of Chemical Physics, Vol 21, 1953, pp. 1087-1092.

Από αυτή την πιθανότητα μετάβασης, είναι προφανές ότι μια κίνηση προς ανώτερη ενεργειακή κατάσταση είναι πιθανή , αλλά η πιθανότητα σχετίζεται αντίστροφα με τη άνοδο και μειώνεται καθώς η θερμοκρασία μειώνεται. Με αυτό τον τρόπο αλγόριθμος προσπαθεί να αποφύγει τον εγκλωβισμό σε τοπικά βέλτιστα και να κατευθυνθεί σε ολικά βέλτιστη λύση.

Το πρόβλημα με τους 13 κόμβους/λιμένες έχει διαμόρφωση όμοια με το πρόβλημα του περιοδεύοντος πωλητή και συνεπώς ο αλγόριθμος Προσομοιωμένης Ανόπτησης το προσεγγίζει με βάση τα παρακάτω χαρακτηριστικά: Διαμόρφωση (configuration) του προβλήματος με βάση τον αλγόριθμο, δημιουργία τυχαίων αναδιοργανώσεων (rearrangements) στη διαμόρφωση, ορισμός της αντικειμενικής συνάρτησης και του χρονοπρόγραμμα ψύξης (cooling schedule).

- Διαμόρφωση: Οι λιμένες ορίζονται ως $i = 1, 2, 3, ..., N$. Η διαμόρφωση του προβλήματος είναι μια κυκλική αντιμετάθεση αριθμών 1,...,n, ώστε να αναπαριστούν την σειρά με την οποία επισκέφθηκε το πλοίο τους λιμένες. Για το υπό έρευνα πρόβλημα το Ν είναι ίσο με 13.

- Αναδιοργάνωση: Οι αναδιοργανώσεις είναι αλλαγές στη σειρά των λιμένων και συνεπώς στη διαδρομή του πλοίου. Δύο λιμάνια επιλέγονται τυχαία και μετά αντιστρέφουμε τη σειρά με την οποία τα επισκέφθηκε το πλοίο. Αυτή η μικρή διαταραχή γίνεται με βάση το Two Bond Exchange[4].

[4] S.Lin and B.Kernigan, An effective heuristic for the traveling salesman problem, Operations Research, Vol 21, 1973, pp 498-516.

- Αντικειμενική Συνάρτηση : Η αντικειμενική συνάρτηση και οι περιορισμοί του προβλήματος περιγράφονται στη εξίσωση (4.1). Για το υπό έρευνα πρόβλημα το κόστος μετάβασης υπολογίζεται από επίσημα στοιχεία του Υ.Ε.Ν.

- Χρονοπρόγραμμα Ψύξης : Αποτελεί το πιο κρίσιμο σημείο της υλοποίησης καθώς έχει πειραματικά δειχθεί ότι τα αποτελέσματα επηρεάζονται σοβαρά από το χρονοπρόγραμμα ψύξης ώστε να μη εγκλωβιστεί ο αλγόριθμος σε τοπικό βέλτιστο. Μετά από πειραματικές προσπάθειες καταλήγουμε ότι το πλέον αρμόζον χρονοπρόγραμμα ψύξης είναι αυτό με αρχική θερμοκρασία ίση με 100, τελική θερμοκρασία ίση με 5000, γραμμικό παράγοντα μείωσης της θερμοκρασίας ίσο με 0,99 και με πλήθος επαναλήψεων ανά θερμοκρασία ίσο με 100N αναδιοργανώσεων (rearrangements) ή 10N επιτυχών αναδιοργανώσεων με N να συμβολίζει το πλήθος των κόμβων του προς μελέτη προβλήματος.

6.4 Η εφαρμογή του Αλγόριθμου του Floyd και του Αλγόριθμου Προσομοιωμένης Ανόπτησης.

Περιορισμοί και Παραδοχές

Οι περιορισμοί αυτής της εφαρμογής τίθενται από το γράφημα των λιμένων που πρέπει να είναι ισχυρά συνεκτικό. Για αυτό χρησιμοποιούνται και ο αλγόριθμος του Floyd και ο αλγόριθμος προσομοιωμένης ανόπτησης, ώστε να προσδιοριστεί το ελάχιστο κόστος μετάβασης για κάθε κόμβο ξεχωριστά και μετά η ελάχιστη συνολική διαδρομή με το αντίστοιχο κόστος.

Οι παραδοχές της εφαρμογής είναι ότι η πιθανότητα αποδοχής μιας νέας λύσης στηρίζεται στο κριτήριο Metropolis. Επίσης, οι νέες λύσεις δημιουργούνται επιλέγοντας τυχαία δυο λιμάνια, w και q, από τη διαδρομή του πλοίου και μετά αντιστρέφουμε τη σειρά με την οποία τα επισκέφθηκε το πλοίο. Ο αλγόριθμος σταματά όταν περεταίρω ελαχιστοποίηση της αντικειμενικής συνάρτησης γίνεται αισθητά πιο δύσκολή με βάση το χρονοπρόγραμμα ψύξης και την καλύτερη λύση μέχρι εκείνη την επανάληψη.

Εφαρμογή του Αλγόριθμου του Floyd

Ο αλγόριθμος του Floyd (μερικές φορές αναφέρεται ως Floyd-Warshall Algorithm), με βάση μία μέθοδο δυναμικού προγραμματισμού βρίσκει την ελάχιστη απόσταση ανάμεσα σε ένα ζεύγος κορυφών ενός σταθμισμένου κατευθυνόμενου γραφήματος. Στο υπό έρευνα πρόβλημα τα λιμάνια αναπαριστούν τις κορυφές και τα τόξα είναι οι μεταξύ τους αποστάσεις, άρα για δύο λιμάνια που συνδέονται με μία διαδρομή ο αλγόριθμος μπορεί να ανακαλύψει την ελάχιστη μεταξύ τους απόσταση. Δηλαδή με είσοδο ένα πίνακα γειτνίασης A, nxn ως έξοδο έχουμε ένα πίνακα C, nxn με τις ελάχιστες αποστάσεις ανά ζεύγος κορυφών. Παρακάτω δίδεται ο ψευδοκώδικας του αλγορίθμου.

/* Θέτουμε τη συνάρτηση edgeCost(i,j) η οποία δίδει το κόστος μετάβασης από το κόμβο i στο j

(άπειρο αν δεν υπάρχει).

Επίσης θέτουμε n τον αριθμό των κορυφών και edgeCost(i,i)=0

*/

```
int path[][];
```

/* Ένας nxn πίνακας. Σε κάθε βήμα του αλγορίθμου, το path[i][j] είναι η ελάχιστη διαδρομή από το i στο j με χρήση ενδιάμεσων τιμών στο (1..k-1). Κάθε path[i][j] αρχικοποιείτε με edgeCost(i,j).

*/

```
Procedure FloydWarshall ()
    for k: = 1 to n
    begin
        for each (i,j) in (1..n)
        begin
            path[i][j] = min ( path[i][j], path[i][k]+path[k][j] );
        end
    end
endproc
```

Εφαρμογή Αλγόριθμου Προσομοιωμένης Ανόπτησης

Σε εφαρμογές του αλγόριθμου Προσομοιωμένης Ανόπτησης, ερευνητές[5] τονίζουν τα ιδιαίτερα χαρακτηριστικά του αλγορίθμου. Παράγοντες όπως S_0, T_0, i_0 ,α,β, που παρουσιάζονται στο ψευδοκώδικα έχουν σημαντική επιρροή στην απόδοση του. Η επιλογή των κατάλληλων τιμών εξαρτάται σε μεγάλο βαθμό από το υπό έρευνα πρόβλημα. Παρακάτω δίδεται ο ψευδοκώδικας του αλγορίθμου.

Procedure SimulatedAnnealing:

{General Form of SimulatedAnnealing}

S:=*S₀*{initial solution}

T:=*T₀*{initial temperature}

iterations :=*i₀*{initial number of iterations of inner loop,>=1}

repeat

repeat

NewS:=perturb (*S*);

if(h(*NewS*)<h(*S*) **or** (random<$e^{(h(S)-h(NewS))/T}$)

then *accept:*=**true**

else *accept:*=**false;**

until inner loop has been repeated *iterations* times;

[5] Nahar S., Sahni S., Shragowitz E., Simulated Annealing and Combinatorial Optimization, in "Proceedings of the 23rd Design Automation Conference", 1986, pp 293-299

*T:=α*T; iterations:=β*iterations*

until "out of time";

end; {of SimulatedAnnealing}

Μια τυχαία αρχική τιμή λύση δημιουργείται και θεωρείται η καλύτερη μέχρι εκείνη τη στιγμή. Ως μέγιστη θερμοκρασία θέτουμε την τρέχουσα θερμοκρασία (Temp(i)), η οποία τίθεται από τον ερευνητή, και ο αριθμός των επιτυχημένων κινήσεων (δηλαδή των αναδιοργανώσεων που οδήγησαν σε νέες λύσεις) για αυτή τη θερμοκρασία τίθεται ίσος με μηδέν (ITRY=0). Με βάση τα παραπάνω η αρχική λύση θεωρείται ως τρέχουσα λύση(current solution) αλλά και λύση γονέας (parent solution). Η λύση γονέας ανανεώνει τη τρέχουσα λύση μόνο όταν κριθεί αναγκαίο.

Ο μετρητής των επιτρεπόμενων κινήσεων αυξάνεται κατά μία μονάδα (ITRY=ITRY+1), εάν η μέγιστη τιμή επιτρεπόμενων κινήσεων για τη συγκεκριμένη θερμοκρασία δεν έχει ξεπεραστεί (ITRY>MaxITRY). Εάν αυτή η τιμή ξεπεραστεί τότε ελέγχεται η τρέχουσα θερμοκρασία. Εάν αυτή είναι ίση με την αρχική (μικρότερη) θερμοκρασία (Temp=Temp(s)), τότε ο αλγόριθμος τερματίζεται, αφού έχει βρεθεί η τελική βέλτιστη λύση (final best solution), κατά τη διάρκεια εκτέλεσής του. Εάν η τρέχουσα θερμοκρασία είναι μεγαλύτερη από την αρχική(Temp>Temp(s)), γίνεται μείωση της θερμοκρασίας με βάση τον παράγοντα μείωσης της θερμοκρασίας και η τιμή επιτρεπόμενων κινήσεων ξαναγίνεται ένα (ITRY=1).

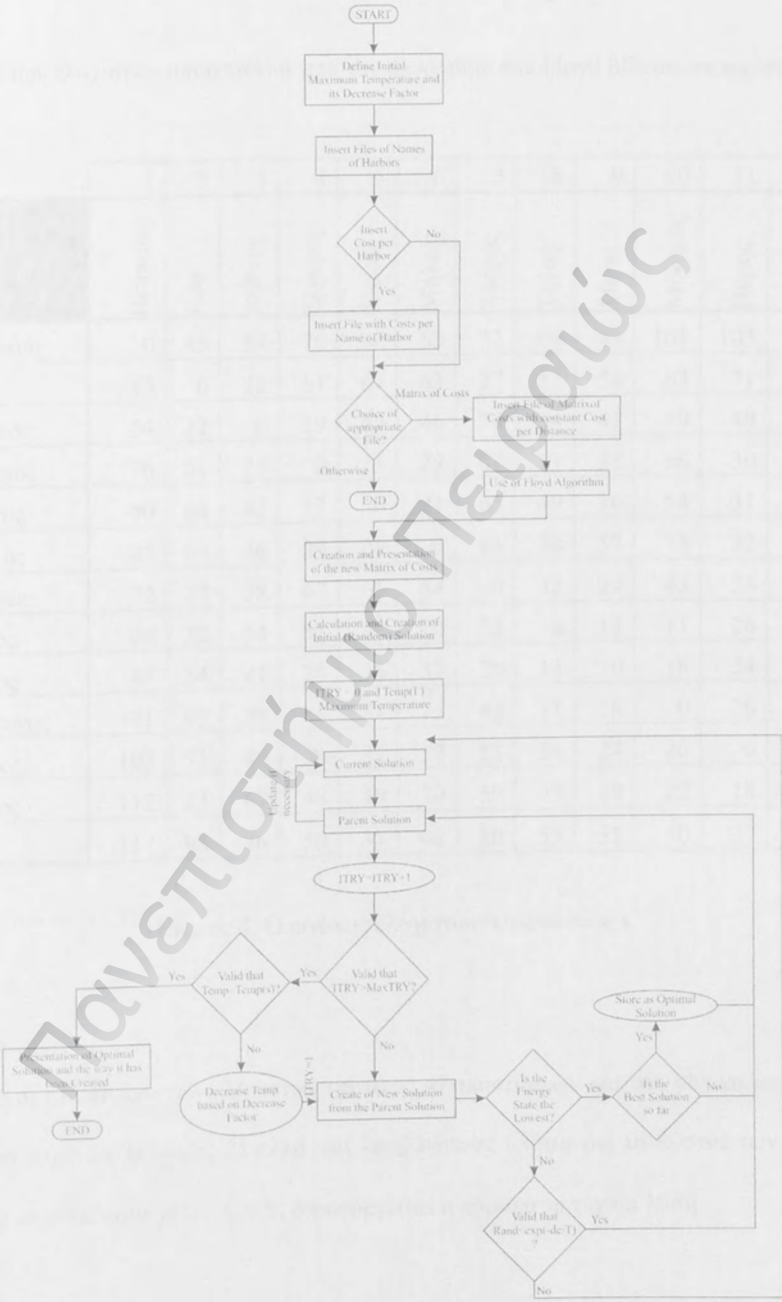
Όταν η μέγιστη τιμή επιτρεπόμενων κινήσεων είναι μικρότερη από το όριο και/ή τρέχουσα θερμοκρασία μειώνεται θα δημιουργηθεί μια νέα λύση με βάση το πρόγραμμα αναδιοργανώσεων του αλγορίθμου.

Εάν η ενέργεια (κόστος) της νέας λύσης είναι μικρότερη της ενέργειας (κόστος) της λύσης γονέας γίνεται έλεγχος έναν αυτή είναι η ολικά καλύτερη και αν είναι αποθηκεύεται. Τελικά η νέα λύση γίνεται η λύση γονέας της επόμενης επανάληψης.

Εάν η ενέργεια (κόστος) της νέας λύσης είναι μεγαλύτερη της ενέργειας (κόστος) της λύσης γονέας κατά ΔΕ, η νέα λύση είναι πιθανό να γίνει αποδεκτή με πιθανότητα

$\Pr\{Accept\Delta E\} = e^{-\Delta E/T}$ με Τ την τρέχουσα θερμοκρασία.

Το διάγραμμα ροής της εφαρμογής δίδεται παρακάτω.



Εικόνα 1: Διάγραμμα Ροής.

Ο πίνακας των ελάχιστων αποστάσεων από τον αλγόριθμο του Floyd δίδεται παρακάτω.

		1	2	3	4	5	6	7	8	9	10	11	12	13
		Πειραιάς	Κέα	Κύθνος	Σέριφος	Σίφνος	Μήλος	Άνδρος	Τήνος	Σύρος	Μύκονος	Πάρος	Νάξος	Ίος
1	Πειραιάς	0	45	54	76	80	85	72	96	83	101	103	112	117
2	Κέα	45	0	22	51	64	63	27	52	54	63	71	83	98
3	Κύθνος	54	22	0	29	42	46	38	54	41	59	49	67	76
4	Σέριφος	76	51	29	0	13	29	67	51	38	56	30	48	50
5	Σίφνος	80	64	42	13	0	21	65	49	36	54	31	49	37
6	Μήλος	85	63	46	29	21	0	84	70	57	75	52	70	48
7	Άνδρος	72	27	38	67	65	84	0	32	29	43	53	58	80
8	Τήνος	96	52	54	51	49	70	32	0	13	11	26	33	53
9	Σύρος	83	54	41	38	36	57	29	13	0	18	24	29	51
10	Μύκονος	101	63	59	56	54	75	43	11	18	0	26	22	50
11	Πάρος	103	71	49	30	31	52	53	26	24	26	0	18	27
12	Νάξος	112	83	67	48	49	70	59	33	29	22	18	0	28
13	Ίος	117	98	76	50	37	48	80	53	51	50	27	28	0

Πίνακας 4: Ο πίνακας ελάχιστων αποστάσεων C

Με τη χρήση του πίνακα των ελάχιστων (πίνακας 4) αποστάσεων και του πίνακα με το κόστος χρήσης των λιμένων (πίνακας 2) αλλά και λαμβάνοντας υπόψη ότι το κόστος των καυσίμων τίθεται ίσο με 5 για κάθε μίλι, $c_f = 5$, δημιουργείται η παρακάτω τυχαία λύση.

Αρχική Τυχαία Λύση				
1	Πειραιάς	→	2	Κέα
2	Κέα	→	3	Κύθνος
3	Κύθνος	→	4	Σέριφος
4	Σέριφος	→	5	Σίφνος
5	Σίφνος	→	6	Μήλος
6	Μήλος	→	7	Άνδρος
7	Άνδρος	→	8	Τήνος
8	Τήνος	→	9	Σύρος
9	Σύρος	→	10	Μύκονος
10	Μύκονος	→	11	Πάρος
11	Πάρος	→	12	Νάξος
12	Νάξος	→	13	Ίος
13	Ίος	→	1	Πειραιάς

Τιμή Αντικειμενικής Συνάρτησης :2676

Πίνακας 5: Αρχική Τυχαία Λύση

Οι τιμές στα τόξα του γραφήματος 2 αναπαριστούν τις ελάχιστες τιμές που προέκυψαν από την εκτέλεση του αλγορίθμου. Ο πίνακας 6 δείχνει αναλυτικά τις τιμές της αντικειμενικής συνάρτησης για τις δέκα επιτυχημένες κινήσεις σε τρείς επαναλήψεις ,δηλαδή νέες λύσεις που προέκυψαν από αναδιοργανώσεις. Όπως αναμένεται καθώς αυξάνονται οι επαναλήψεις μειώνεται η αντικειμενική συνάρτηση. Αυτό συμβαίνει καθώς μετά από το πέρας κάθε επανάληψης ο αλγόριθμος τείνει σε καλύτερη λύση και τελικά συγκλίνει προς την τελική λύση.

	Επιτυχημένες Κινήσεις						
Επαναλήψη	1	2	3	4	5	6	7
1	2676	2651	2636	2271	2206	2181	2156
2	2131	2201	-	-	-	-	-
3	2131	-	-	-	-	-	-

Πίνακας 6

Η πρώτη επιτυχημένη κίνηση στην επανάληψη 2 έχει την ίδια τιμή στην αντικειμενική συνάρτηση με τη πρώτη επιτυχημένη κίνηση στην επανάληψη 3 αν και προκύπτουν από διαφορετικές διαδρομές. Η διαδρομή δείχνεται στο πίνακα 7.

Επανάληψη 2 - Κίνηση 1				
8	Τήνος	→	10	Μύκονος
10	Μύκονος	→	12	Νάξος
12	Νάξος	→	11	Πάρος
11	Πάρος	→	13	Ίος
13	Ίος	→	6	Μήλος
6	Μήλος	→	5	Σίφνος
5	Σίφνος	→	4	Σέριφος
4	Σέριφος	→	3	Κύθνος
3	Κύθνος	→	1	Πειραιάς
1	Πειραιάς	→	2	Κέα
2	Κέα	→	7	Άνδρος
7	Άνδρος	→	9	Σύρος
9	Σύρος	→	8	Τήνος

Τιμή Αντικειμενικής Συνάρτησης :2131

Πίνακας 7

Επιτυγχάνουμε το ακόλουθο αποτέλεσμα ως τελική λύση μετά από δέκα επιτυχημένες κινήσεις και τρείς επαναλήψεις

Τελική Λύση				
1	Πειραιάς	→	3	Κύθνος
3	Κύθνος	→	4	Σέριφος
4	Σέριφος	→	5	Σίφνος
5	Σίφνος	→	6	Μήλος
6	Μήλος	→	13	Ίος
13	Ίος	→	11	Πάρος
11	Πάρος	→	12	Νάξος
12	Νάξος	→	10	Μύκονος
10	Μύκονος	→	8	Τήνος
8	Τήνος	→	9	Σύρος
9	Σύρος	→	7	Άνδρος
7	Άνδρος	→	2	Κέα
2	Κέα	→	1	Πειραιάς

Τιμή Αντικειμενικής Συνάρτησης :2131

Πίνακας 8: Τελική Λύση

Το γράφημα της τελικής λύσης δίδεται παρακάτω με τις τιμές της αντικειμενικής συνάρτησης ανά κόμβο/λιμένα.

Γράφημα 2: Η τελική λύση με τις τιμές της αντικειμενικής συνάρτησης ανά κόμβο/λιμένα.

6.5 Συμπεράσματα

Η προσπάθεια να ελαχιστοποιηθεί το συνολικό κόστος μετάβαση για δεκατρία λιμάνια, λαμβάνοντας υπόψη το κόστος των καυσίμων και χρήσης των λιμανιών, με τη χρήση των δύο αλγορίθμων δείχνει την ικανότητα της πρότασης να ανακαλύπτει γρήγορα ποιοτικές λύσεις. Επίσης το εγγενές χαρακτηριστικό της αποφυγής εγκλωβισμού σε τοπικά βέλτιστα, καθώς είναι δυνατό να μεταβεί και σε χειρότερη λύση σε κάποια επανάληψη, τον διαφοροποιεί σε σχέση με άλλους αλγορίθμους. Ο κύριος ρυθμιστικός παράγοντας της συμπεριφοράς του αλγορίθμου είναι το χρονοπρόγραμμα ψύξης. Ένα σχετικά γρήγορο χρονοπρόγραμμα ψύξης, όπως και στο φυσικό μοντέλο ανόπτησης μετάλλων, δεν θα δώσει στα μόρια/τμήματα λύσης το χρόνο που χρειάζονται ώστε να διαταχθούν με την ελάχιστη ενέργεια/κόστος. Όμως, ένα πολύ αργό πρόγραμμα ψύξης θα αυξήσει δυσανάλογα τον υπολογιστικό χρόνο που χρειάζεται για την παραγωγή λύσεων. Σημαντικό επίσης είναι το κριτήριο αποδοχής νέων λύσεων καθώς σε αυτό το σημείο κρίνεται η επιλογή μεταξύ εξερεύνησης νέων λύσεων και εκμετάλλευσης παλαιότερων. Αποτελέσματα υψηλής ποιότητας βασίζονται στη σωστή επιλογή αυτών των κριτηρίων στο υπό έρευνα πρόβλημα και ουσιαστικά από αυτά εξαρτάται η ευρωστία του αλγορίθμου.

Κεφάλαιο 7

Εφαρμογές του Αλγορίθμου Ant Colony System

7.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται εφαρμογές[1] του αλγόριθμου Ant Colony System. Οι εφαρμογές του αλγορίθμου αφορούν σε πρόβλημα δρομολόγησης οχημάτων, πρόβλημα βέλτιστης διαχείρισης θαλάσσιων μεταφορών καθώς και πρόβλημα διάταξης μηχανών σε κατασκευαστική κυψέλη.

Το κλασικό Πρόβλημα Δρομολόγησης Οχημάτων ορίζεται μέσω ενός πλήρους γραφήματος $G=(N,A)$, όπου:

$N=(0,1,2,...,n)$ είναι το σύνολο των κόμβων ενώ το 0 εκφράζει την κεντρική βάση

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

Σκοπός είναι να ανακαλύψουμε διαδρομές ελάχιστου κόστους έτσι ώστε (i) να επισκεφθούμε κάθε πελάτη ακριβώς μία φορά από ακριβώς μόνο ένα όχημα, (ii) για κάθε όχημα η ζήτηση της διαδρομής δεν υπερβαίνει τη χωρητικότητα D (iii) η συνολική διαδρομή δεν υπερβαίνει ένα όριο L και (iv) κάθε όχημα ξεκινά και τερματίζει τη διαδρομή του στην αφετηρία (depot).

Το Πρόβλημα Δρομολόγησης Οχημάτων μπορεί να μετατραπεί σε ένα Πρόβλημα Περιοδεύοντος Πωλητή αν χωρίσουμε τους πελάτες/κόμβους σε N υποσύνολα τα οποία εξυπηρετούνται από ένα όχημα απεριόριστής χωρητικότητας. Στο Πρόβλημα του Περιοδεύοντος

[1] Βλάχος Α., (2005), Μέτα-ευριστικοί Αλγόριθμοι Βελτιστοποίησης και Συνδυαστικά Προβλήματα Βελτιστοποίησης, Διδακτορική Διατριβή, Τμήμα Πληροφορικής, Πανεπιστήμιο Πειραιώς.

Πωλητή από μια αφετηρία ξεκινά ο πωλητής ο οποίος αφού επισκεφθεί ένα σύνολο πελατών επιστρέφει σε αυτή. Άρα για Ν σύνολα πελατών, λύνουμε Ν προβλήματα περιοδεύοντος πωλητή, το οποίο είναι γνωστό ότι αυξάνει σε πολυπλοκότητα εκθετικά ως προς τον αριθμό των κόμβων/πελατών. Έτσι βλέπουμε ότι το Πρόβλημα Δρομολόγησης Οχημάτων σχετίζεται στενά με το Πρόβλημα του Περιοδεύοντος Πωλητή καθώς είναι σαν να επιλύουμε πολλαπλά προβλήματα περιοδεύοντος πωλητή με κοινή πόλη αφετηρία και τερματισμό.

Η μορφολογία της χώρας μας με το μεγάλο αριθμό νησιών δημιουργεί πλήθος προβλημάτων δρομολόγησης στις θαλάσσιες μεταφορές. Η διεύθυνση μεταφορών της Ευρωπαϊκής Ένωσης[2] μελετά την μεταφορά αγαθών με πλοία εμπορευματοκιβωτίων για το ενδοκοινοτικό εμπόριο με σκοπό τη βέλτιστη διαχείριση του δικτύου αλλά και της εξυπηρέτησης των πολιτών. Παρακάτω στο κεφάλαιο εξετάζουμε ένα δίκτυο θαλάσσιων μεταφορών με δεκατρείς κόμβους στο Αιγαίο με σκοπό την βέλτιστη διαχείριση του. Παρακάτω παρουσιάζονται η έκφραση του προβλήματος, η ανάπτυξη των παραδοχών και των βημάτων του αλγορίθμου που χρησιμοποιείται και τελικά μελετάται η επίδραση των μεταβολών της αρχικής ποσότητας φερομόνης στην ελαχιστοποίηση της αντικειμενικής συνάρτησης του προβλήματος.

Η Κυψελοειδής Διαδικασία Παραγωγής αναφέρεται σε ένα σύστημα παραγωγής στο οποίο τα μηχανήματα παραγωγής είναι διατεταγμένα με σκοπό την συνεχή αποτελεσματική ακολουθία των υλικών έτσι ώστε να παραχθούν τα προϊόντα από την αρχή μέχρι το τέλος με μια διαδικασία ροής, ενώ ελαχιστοποιούνται οι μετακινήσεις και ο χρόνος αναμονής. Για την δημιουργία μιας διαδικασίας ροής πρέπει να αναγνωριστούν οι μηχανές που θα χρησιμοποιηθούν για την παραγωγής της οικογένειας των προϊόντων και να τοποθετηθούν σε ένα κοινό χώρο.

---

[2] European Cooperation in the Field of Scientific and Technical Research (COST). Cost 339 small containers. Final report of the Action. Energy and Transport Directorate General. European Commission, 2000.

Το μοντέλο της ελάχιστης προς τα πίσω ροής με χρήση σαφώς ορισμένων δεδομένων για την παραγωγή είναι σχετικά εύκολο προσομοιωθεί. Τα πιο σημαντικά δεδομένα είναι η σαφώς ορισμένα όπως η διεργασία παραγωγής και η ζητούμενη ποσότητα παραγωγής, αλλά και πιθανές μεταβολές αυτών που θα προκαλέσουν τροποποίηση της διάταξης των μηχανών. Το μοντέλο υιοθετεί έναν αριθμό από περιορισμούς με στόχο την προσέγγιση της βέλτιστης ακολουθίας μηχανών που ελαχιστοποιεί τις προς τα πίσω ροές στην κυψέλη παραγωγής όταν η γραμμή παραγωγής είναι σε σειρά.

Η εύρεση της βέλτιστης τοποθέτησης για το πρόβλημα τοποθέτησης μηχανών σε σειρά με τη χρήση του αλγόριθμου Ant Colony System είναι ο ερευνητικός σκοπός και ενώ το πλαίσιο των Αλγορίθμων Μυρμηγκιών είναι ευπροσάρμοστο, η υλοποίηση του αλγορίθμου πρέπει να λαμβάνει υπόψη τους περιορισμούς του προβλήματος ώστε να παραχθούν ποιοτικά αποτελέσματα. Συνεπώς, όπως σε όλες τις εφαρμογές του πλαισίου ACO, ο σωστός σχεδιασμός του αλγορίθμου και η ρύθμιση των παραμέτρων σε συνδυασμό με το υπό έρευνα πρόβλημα θα χρίζουν ιδιαίτερης προσοχής.

Ψευδοκώδικας Αλγορίθμου

Παρακάτω παρουσιάζονται οι προϋποθέσεις που ισχύουν και στη συνέχεια δίδεται ο ψευδοκώδικας του αλγόριθμου που χρησιμοποιήθηκε για την επίλυση προβλημάτων δρομολόγησης και βελτιστοποίησης θαλάσσιων μεταφορών. Θεωρούμε το σύνολο των οχημάτων/μυρμηγκιών και ορίζουμε τη μεταβλητή LoadperAnt (Q_k) η οποία δίδει το φορτίο κάθε ένα και ενημερώνεται μετά από κάθε κόμβο που επισκέπτεται. Οποιοδήποτε μυρμήγκι μπορεί να μετακινηθεί σε οποιοδήποτε κόμβο, εφόσον σ' αυτό υπάρχει ζήτηση και αν η ζήτηση του κόμβου έχει πληρωθεί, τότε ο κόμβος αγνοείται από τα οχήματα/μυρμήγκια αλλά χρησιμοποιείται ως ενδιάμεσος σταθμός για τη μετάβαση σε κόμβο που υπάρχει ζήτηση.

Όταν ένα όχημα επισκέπτεται ένα κόμβο, το όχημα αυτό θα πρέπει να δώσει όλο του το φορτίο έως η ζήτηση του κόμβου να πληρωθεί. Όχημα δίχως φορτίο επιστρέφει στην αφετηρία, αλλά το κόστος επιστροφής δεν υπολογίζεται και μετά την επιστροφή στην αφετηρία ανανεώνεται και η μνήμη του οχήματος/μυρμηγκιού (AM) ώστε να μπορεί να ξαναεπισκεφθεί όλους του κόμβους που έχουν ακόμα απαιτήσεις. Με την πλήρωση των απαιτήσεων όλων των κόμβων, όλα τα οχήματα επιστρέφουν στην αφετηρία και επαναϋπολογίζεται η ποσότητα της φερομόνης στις διαδρομές με βάση την χρήση τους.

Procedure –ACS for Transport Problems

$S := S_0$ {initial solution}

$\tau = \tau_0$ {initial pheromone}

iterations:=i0 {initial number of iterations>=1}

Read from file

NN←number of nodes

ArcSet←node (x),destination node (y),route cost (c),node demand (d)

Initialize

Set initial pheromone value for all arcs (τ_0)

Set iteration number = termination condition

Set load value per ant

Set AM←Ant Memory

```
while nodedemand<Full do
    Route ants to that destination
    Increment nodedemand counter
    while terminationcondition not met do
        for each ant k ∈ {1,2,...,m} do
            if loadvalueperant =0 then
                Ant returns to starting point
            else
                Repeat for each node j
                if node (j-1,j) exists, nodedemand<Full, j ≠ AM then
                    Calculate transition probability for ant k
                    Ant k selects next node based on transition probability
                endif
            endif
        endfor
        Calculate objective function value
        Calculate pheromone value for each ant
    endwhile
endwhile
end-procedure ACS for Transport Problems
```

7.2 Πρόβλημα Δρομολόγησης Οχημάτων

7.2.1 Έκφραση του Προβλήματος

Έστω $G=(N,A,d)$ ένα πλήρες σταθμισμένο κατευθυνόμενο γράφημα, όπου $N=(n_0,n_1,....,n_n)$ το σύνολο των κόμβων, $A=\{(i,j): i\neq j\}$ το σύνολο των τόξων και για κάθε τόξο (i,j) υπάρχει το αντίστοιχο βάρος $d_{ij}\geq 0$ που είναι η απόσταση μεταξύ του n_i και του n_j. Ο κόμβος n_0 είναι η αφετηρία (depot) όπου M οχήματα υπάρχουν και το καθένα έχει χωρητικότητα D, και οι υπόλοιποι κόμβοι είναι οι πελάτες που πρέπει να εξυπηρετηθούν.

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.

Το ζητούμενο είναι ο καλύτερος συνδυασμός διαδρομών για ένα ορισμένο αριθμό οχημάτων, ώστε να ικανοποιούνται οι απαιτήσεις ενός, επίσης ορισμένου αριθμού κόμβων, οι οποίοι ενώνονται μεταξύ τους με διαδρομές διαφορετικού κόστους. Για την εφαρμογή αλγορίθμου Ant Colony System[3] (ACS) κάθε μυρμήγκι αντιστοιχεί σε ένα όχημα. Ιδιαίτερα σημαντικός είναι ο τρόπος με τον οποίον το μυρμήγκι επιλέγει τον επόμενο κόμβο καθώς η επιλογή σχετίζεται με την ποσότητα της φερομόνης που εναποτίθεται σε κάθε τόξο και με το κόστος διάσχισης αντίστοιχα.

Στο τέλος της κάθε επανάληψης του αλγορίθμου, υπολογίζεται η τιμή της αντικειμενικής συνάρτησης του προβλήματος δρομολόγησης οχημάτων η οποία δίνεται από την εξίσωση:

[3] Dorigo, M., Gambardella, L.M., (1997) Ant Colony System: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation, 1 (1), p. 53-66

$$f_{obj} = \sum \mu_{ij} \cdot C_{ij}^{-1} \tag{7.4}$$

όπου μ_{ij} αριθμός των διασχίσεων που έγιναν για το τόξο που ενώνει τους κόμβους i και j

$C_{i,j}^{-1} = L_{ij} C_{L} + C_{pj}$ κόστος διάσχισης για το τόξο που ενώνει τους κόμβους i και j

Η αντικειμενικής συνάρτηση υπολογίζεται αθροίζοντας το κόστος διάσχισης κάθε τόξου που χρησιμοποιήθηκε, επί τον αριθμό των φορών που χρησιμοποιήθηκε.. Η παράμετρος β ορίζεται από το χρήστη όπως και η παράμετρος ρ, η εξάτμιση της φερομόνης.

7.2.2 Εφαρμογή του Ant Colony System για το Πρόβλημα Δρομολόγησης Οχημάτων

Οι τρεις παράμετροι που πρέπει να οριστούν πριν από την εκκίνηση της εφαρμογής του αλγορίθμου είναι:

- το μέγιστο φορτίο που μπορεί να μεταφέρει ο κάθε πράκτορας (μυρμήγκι)
- το πλήθος των επαναλήψεων του αλγορίθμου 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 κάθε επανάληψης. Το πλήθος των επαναλήψεων συναρτάται με την ποιότητα των αποτελεσμάτων που θα προκύψουν από τη λειτουργία του αλγορίθμου. Η αρχική ποσότητα φερομόνης πρέπει να επιλέγεται σε σχέση με το υπό έρευνα πρόβλημα και με προσοχή καθώς μετά το πέρας μιας επανάληψης, εφαρμόζεται ο κανόνας ανανέωσης της φερομόνης του ACS. Αυτό σημαίνει ότι η ποσότητα της φερομόνης σε κάθε τόξο στην επόμενη επανάληψη θα ενισχυθεί με τρόπο τέτοιο ώστε στην επόμενη επανάληψη, τα μυρμήγκια/οχήματα θα προτιμήσουν τόξα με μικρότερο κόστος και η διαδικασία θα συνεχίζεται καθώς αυξάνει ο αριθμός των επαναλήψεων. Ο αλγόριθμος υλοποιήθηκε για ένα γράφημα οκτώ (8) κόμβων με αντίστοιχες απαιτήσεις και για αρχικές ποσότητες φερομόνης όπως δείχνεται στον πίνακα 1 και πίνακα 2 με β=1 και ρ=0.5.

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

Πίνακας 1

τ_0	Q_k
500	50
4000	50
50	50

Πίνακας 2

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 επαναλήψεις του αλγορίθμου.

Σχήμα 7.1: Επίδραση μεταβολής της αρχικής ποσότητας φερομόνης στη βελτιστοποίηση της αντικειμενικής συνάρτησης του προβλήματος δρομολόγησης οχημάτων.

Από το σχήμα 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, ενώ εμπειρικά γνωρίζουμε ότι υπάρχουν περιθώρια βελτίωσης. Δηλαδή μια σχετικά μεγάλη αρχική τιμή φερομόνης εγκλώβισε την αποικία σε τοπικό βέλτιστο.

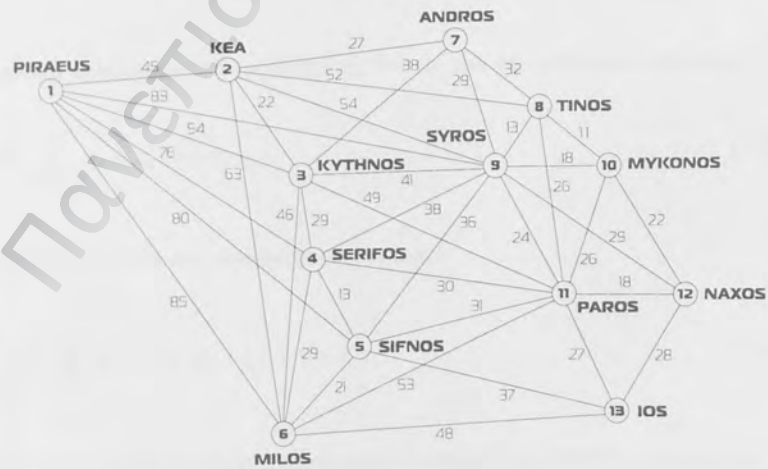
Για $\tau_0 = 50$, παρατηρείται πως το αποτέλεσμα της αντικειμενικής συνάρτησης στο τέλος της πρώτης επανάληψης είναι χαμηλότερο απ' ότι στις άλλες δύο περιπτώσεις, αλλά παρατηρείται πολύ μικρή ή καθόλου βελτίωση στο τέλος των υπολοίπων επαναλήψεων.

Για $\tau_0 = 500$ προκύπτουν τα καλύτερα αποτελέσματα. Παρατηρείται μια πολύ καλή βελτιστοποίηση της αρχικής τιμής με την αντικειμενική συνάρτηση να μειώνεται με πολύ γρήγορο ρυθμό και φτάνει σε μια σχεδόν σταθερή κατάσταση στο τέλος της δέκατης επανάληψης.

7.3 Βελτιστοποίηση Θαλάσσιων Μεταφορών

7.3.1 Έκφραση του Προβλήματος

Στο σχήμα 1 απεικονίζετε τα δεκατρία νησιά κόμβοι που πρέπει να εξυπηρετηθούν από τη χωρητικότητα ενός στολίσκου πλοίων εμπορευματοκιβωτίων με βάση το λιμάνι του Πειραιά. Οι συνδέσεις ανάμεσα στους κόμβους του σχήματος απεικονίζουν υπαρκτές ή εφικτές θαλάσσιες διαδρομές ανάμεσα σε αυτά τα νησιά. Η απαίτηση του κάθε κόμβου για εμπορευματοκιβώτια είναι σαφώς ορισμένη και με αφετηρία τον Πειραιά θα εξυπηρετηθεί από τα πλοία του στολίσκου τα οποία έχουν πλήθος διαδρομών να επιλέξουν με διαφορετικό κόστος. Το κόστος κάθε θαλάσσιας διαδρομής υπολογίζεται ανά ναυτικό μίλι καθώς κάθε πλοίο θεωρείται ότι καταναλώνει ποσότητα καυσίμου ανάλογα με την απόσταση. Επίσης στο κόστος πρέπει να προσθέσουμε το κόστος ελλιμενισμού το οποίο μεταβάλλεται σε κάθε νησί. Για να μειωθεί η πολυπλοκότητα του προβλήματος, το κόστος επιστροφής στο λιμάνι του Πειραιά δεν υπολογίζεται.



Σχήμα 1: Το επιλεγμένο σύνολο των 13 κόμβων/λιμένων και οι αποστάσεις τους.

Οι παρακάτω μεταβλητές χρησιμοποιούνται στις εξισώσεις

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 ανά ναυτικό μίλι

c_{p_i} το κόστος χρήσης του λιμένα i

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 κόμβο j

D_i η απαίτηση του λιμένα i (αριθμός των containers)

S_i η παροχή στο λιμένα i (αριθμός των containers)

Η αντικειμενική συνάρτηση του προβλήματος δίδεται από την παρακάτω εξίσωση.

$$\min \sum_{all\ arcs\ ij} n_{ij} \cdot L_{ij} \cdot c_f + n_{ij} \cdot c_{p_i} \tag{7.5}$$

Επίσης λαμβάνονται υπόψη οι παρακάτω περιορισμοί:

$$\sum_{j=1}^{J} W_{ij} - \sum_{k=1}^{K} W_{ik} + S_i - D_i = 0 \quad , \forall\ i, j \tag{7.6}$$

Το ισοζύγιο των εισερχομένων και εξερχομένων container στον κόμβο i είναι μηδέν.

$$n_{ij}Q - W_{ij} \geq 0 \quad , \forall\, i, j \qquad (7.7)$$

Ορίζοντας τον αριθμό των πλοίων n_{ij} επί της χωρητικότητάς τους Q ως μεγαλύτερο ή ίσο με τον αριθμό των containers που μεταφέρθηκαν W_{ij}, δίνεται ένα άνω φράγμα για το W_{ij}.

$$W_{ij} - (n_{ij} - 1)Q \geq 0 \quad , \forall\, i, j \qquad (7.8)$$

Δίδει το κάτω φράγμα του W_{ij}.

$$W_{ij} \geq n_{ij} \quad , \forall\, i, j \qquad (7.9)$$

Διασφαλίζει ότι το W_{ij} παίρνει θετική τιμή όταν το n_{ij} γίνεται μεγαλύτερο του μηδενός.

$$n_{ij} \leq n_{\max} \quad , \forall\, i, j \qquad (7.10)$$

Ορίζει το άνω όριο του n_{ij}, αλλιώς η τιμή του n_{ij} είναι χωρίς όρια και το πρόβλημα δεν μπορεί να λυθεί.

$$L_{ij} = L_{ji} \quad , \forall\, i, j \qquad (7.11)$$

Παραδοχή ότι το τόξο (i, j) έχει ίσο μήκος με το τόξο (j, i).

$$n_{ij} \geq 0 \quad , \forall\, i, j \qquad (7.12)$$

$$W_{ij} \geq 0 \quad , \forall\, i, j \qquad (7.13)$$

Περιορισμός μη αρνητικότητας, και n_{ij} ακέραιος.

7.3.2 Εφαρμογή του Ant Colony System για τη Βελτιστοποίηση Θαλάσσιων Μεταφορών

Σκοπός όπως προαναφέρθηκε είναι η εύρεση της καλύτερης διαδρομής με ελάχιστο μήκος και ελάχιστα οχήματα/πλοία και παράλληλα να ικανοποιηθεί η απαίτηση του κάθε κόμβου/νησιού. Για την εφαρμογή της νέας προσέγγισης του αλγορίθμου Ant Colony System κάθε μυρμήγκι αντιστοιχεί σε ένα πλοίο. Θεωρούμε το σύνολο των πλοίων/μυρμηγκιών και ορίζουμε τη μεταβλητή LoadperAnt (Q_k) η οποία δίδει το φορτίο κάθε ένα και ενημερώνεται μετά από κάθε κόμβο που επισκέπτεται. Όπως στις κλασικές εφαρμογές του ACS κάθε μυρμήγκι (πλοίο) ξεκινάει από έναν ορισμένο κόμβο και ως αφετηρία (depot) θεωρούμε το λιμάνι του Πειραιά.

Αν σε κάποιο κόμβο υπάρχει ζήτηση τότε αυτός μπορεί να εξυπηρετηθεί από οποιοδήποτε μυρμήγκι και αν η ζήτηση του κόμβου έχει πληρωθεί, τότε ο κόμβος αγνοείται από τα μυρμήγκια (πλοία) αλλά χρησιμοποιείται ως ενδιάμεσος σταθμός για τη μετάβαση σε κόμβο που υπάρχει ζήτηση. Η ζήτηση του κόμβου θα πρέπει να ικανοποιηθεί πλήρως ή να δοθεί όλη η ποσότητα που διαθέτει το μυρμήγκι (πλοίο). Μυρμήγκια με μηδενικό φορτίο επιστρέφουν στην αφετηρία, αλλά το κόστος επιστροφής δεν υπολογίζεται και μετά την επιστροφή στην αφετηρία ανανεώνεται και η μνήμη του μυρμηγκιού/πλοίου (AM) ώστε να μπορεί να ξαναεπισκεφθεί όλους του κόμβους που έχουν ακόμα απαιτήσεις.

Με την πλήρωση των απαιτήσεων όλων των κόμβων, όλα τα μυρμήγκια επιστρέφουν στην αφετηρία και επαναϋπολογίζεται η ποσότητα της φερομόνης στις διαδρομές με βάση την χρήση τους.

Η βέλτιστη διαχείριση ενός στολίσκου πλοίων εμπορευματοκιβωτίων είναι ο σκοπός της υλοποίησης. Το σχήμα 1 δίδει γραφικά τις συνδέσεις του δικτύου που συνδέει δεκατρία νησιά κόμβους και βασίζεται σε πληροφορίες από ναυτιλιακούς χάρτες. Οι κόμβοι εξυπηρετούνται από την χωρητικότητα του στόλου που ορίζεται σε 300 εμπορευματοκιβώτια των 1500 τόνων και ως αφετηρία έχει το λιμάνι του Πειραιά. Η ζήτηση των νησιών κόμβων ορίζεται από στοιχεία του Υπουργείου Εμπορικής Ναυτιλίας και απεικονίζεται σε τόνους και με τα στοιχεία αυτά δημιουργούμε τον πίνακα 1. Το μέγεθος του στόλου σχετίζεται με τη χωρητικότητα των εμπορευματοκιβωτίων, καθώς πρέπει να ικανοποιηθούν πλήρως οι απαιτήσεις των κόμβων πελατών, αυτά τα δεδομένα είναι όμως ορισμένα σαφώς από το πρόβλημα. Συνεπώς κατά την εκτέλεση του αλγορίθμου οδηγούμαστε σε κινήσεις πλοίων προς κόμβους, ενώ αυτά είναι χωρίς φορτίο, διότι θα χρησιμοποιηθούν για την ικανοποίηση κόμβων που βρίσκονται στη συνέχεια της διαδρομής. Η ελαχιστοποίηση των κινήσεων των κενών πλοίων και του μεγέθους του στόλου είναι μια άσκηση ισορροπίας καθώς με το γίνουν ορισμένες κινήσεις κενών τελικά θα χρειαστούν λιγότερα πλοία για το σύνολο των κινήσεων.

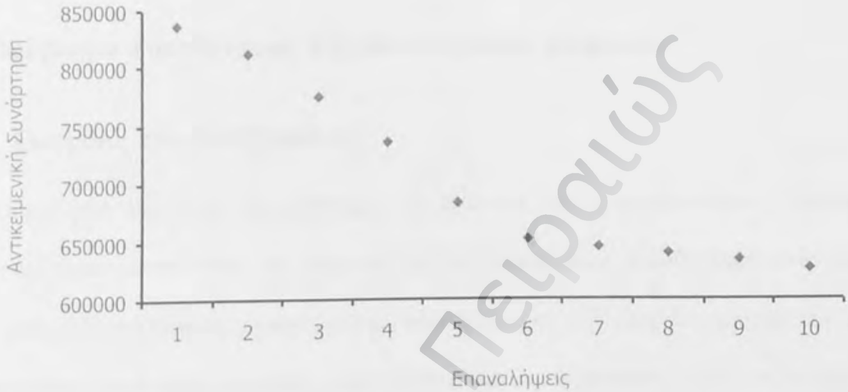
		PORT DEMAND small containers		PORT SUPPLY small containers
1 Πειραιάς	D_1	0	S_1	331368
2 Κέα	D_2	8002	S_2	0
3 Κύθνος	D_3	1684	S_3	0
4 Σέριφος	D_4	2992	S_4	0
5 Σίφνος	D_5	6805	S_5	0
6 Μήλος	D_6	15875	S_6	0
7 Άνδρος	D_7	1847	S_7	0
8 Τήνος	D_8	26495	S_8	0
9 Σύρος	D_9	29477	S_9	0
10 Μύκονος	D_{10}	48805	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**

Πίνακας 1: Απαιτήσεις κόμβων

	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

Πίνακας 2: Αποτελέσματα διαδρομών

Η ελαχιστοποίηση της αντικειμενικής συνάρτησης παρουσιάζεται στο σχήμα 2.



Σχήμα 2 Ελαχιστοποίηση της αντικειμενικής συνάρτησης

Με την έναρξη του ο αλγόριθμος δημιουργεί μια πρώτη λύση στο πρόβλημα. Τα πλοία/μυρμήγκια ξεκινούν από τη αφετηρία, το λιμάνι του Πειραιά, και επιλέγουν με στοχαστικό τρόπο τον επόμενο κόμβο με βάση τον πιθανοκρατικό κανόνα μετάβασής. Όταν ικανοποιηθούν οι απαιτήσεις του προβλήματος, υπολογίζεται το κόστος της διαδρομής και εναποτίθεται φερομόνη στα τόξα που χρησιμοποιήθηκαν με βάση τη διαφοροποιημένη διαδικασία ανανέωσης της φερομόνης.

Η ενημέρωση των τόξων του γραφήματος που χρησιμοποιήθηκαν με τη νέα ποσότητα φερομόνης, γίνεται ανάλογα με το πλήθος των διασχίσεων του και με το κόστος του. Δηλαδή η πιθανότητα να επιλεγεί ένας κόμβος εξαρτάται από την φερομόνη και το μήκος του. Όσο περισσότερη φερομόνη και μικρότερο το μήκος τόσο μεγαλύτερη και η πιθανότητα να επιλεγεί ως ο επόμενος κόμβος.

Ο αλγόριθμος επαναλαμβάνεται με τα νέα δεδομένα φερομόνης που έχουν προκύψει από την προηγούμενη επανάληψη. Ύστερα από δέκα επαλείψεις του αλγορίθμου, οι διαδοχικές τιμές που απέκτησε η αντικειμενική συνάρτηση του προβλήματος παρουσιάζονται στο σχήμα 2.

7.4 Πρόβλημα Τοποθέτησης Κατασκευαστικών Κυψελών

7.4.1 Έκφραση του Προβλήματος

Στη κυψελοειδή παραγωγή το πρόβλημα της διάταξης των μηχανών είναι η εύρεση της καλύτερης τοποθέτησης τους στο χώρο και οι πιο διαδεδομένες τοποθετήσεις είναι τρείς. Σε μονή γραμμή, σε πολλαπλές γραμμές και σε διάταξη βρόχου. Οι ροές που προκύπτουν από μια τέτοια διάταξη είναι προς τα εμπρός, προς τα πίσω και σε παράκαμψη. Οι δύο τελευταίες είναι και οι πλέον ανεπιθύμητες καθώς διακόπτουν τη διαδοχική ροή των ημιτελών προϊόντων από τη μια μηχανή στην άλλη. Η προς τα πίσω ροή συμβαίνει όταν ένα εξάρτημα/προϊόν πρέπει να μετακινηθεί από τη μηχανή που βρίσκεται σε μια άλλη προηγούμενη μηχανή στη γραμμή παραγωγής. Αυτό μπορεί να απαιτείται πολλές φορές, καθώς ορίζεται από την ακολουθία παραγωγής του κάθε εξαρτήματος/προϊόντος ξεχωριστά και είναι η κίνηση που πρέπει να ελαχιστοποιηθεί στο μοντέλο. Στη παρούσα εργασία μελετάται η διάταξη μονής γραμμής όπου τα προϊόντα που παράγονται χρησιμοποιούν την ίδια ακολουθία μηχανών και το προϊόν κινείται σε μια γραμμή ροής.

Η Διάταξη Μονής Σειράς

Η Διάταξη Μονής Σειράς ορίζεται ως η τοποθέτηση κατά την οποία οι μηχανές που επεξεργάζονται τα υλικά είναι η μια δίπλα στην άλλη και ο τρόπος μετακίνησης των υλικών μεταξύ τους γίνεται σε μια νοητή ευθεία. Η μετακίνηση των υλικών σε τέτοιου είδους διατάξεις γίνεται συνήθως με ένα σύστημα ιμάντα ή με Αυτόματα Καθοδηγούμενο Όχημα (Automated Guided Vehicle).

Στη παραγωγή δυστυχώς υπάρχουν προϊόντα που για να κατασκευαστούν δε θα περάσουν με ροή μόνο προς τα εμπρός σε ακολουθία. Ορισμένα πρέπει να μετακινηθούν προς τα πίσω από τον ιμάντα ή το AGV και συνεπώς δημιουργούν ροή προς τα πίσω που είναι ανεπιθύμητη. Σε αυτή την περίπτωση σκοπός είναι η ελαχιστοποίηση της προς τα πίσω ροής (backward flow) και παρουσιάζεται στη συνάρτηση παρακάτω.

$$Obj = \min \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij} d_{ij} \tag{8.1}$$

όπου:

f_{ij} : αριθμός της συχνότητας/όγκο των προϊόντων που μετακινούνται προς τη μηχανή j

d_{ij} :η προς τα πίσω απόσταση μεταξύ της μηχανής j και της μηχανής i

Ο πίνακας f_{ij} δείχνει τις κινήσεις από και προς την κάθε μια μηχανή και λαμβάνεται από τις πληροφορίες που ορίζουν την ακολουθία μηχανών που πρέπει να υλοποιηθεί για την παραγωγή κάθε αγαθού. Η παραγωγή νέου αγαθού γίνεται με διαφορετική ακολουθία μηχανών και συνεπώς δημιουργεί και νέο πίνακα.

Το μοντέλο της Ελαχίστης Προς τα Πίσω Ροής

Το μοντέλο της Ελαχίστης Προς τα Πίσω Ροής έχει ως σκοπό τον προσδιορισμό της ακολουθίας μηχανών που ελαχιστοποιεί τις προς τα πίσω ροές στην κυψέλη παραγωγής και υιοθετεί έναν αριθμό από περιορισμούς. Σύμφωνα με το μοντέλο, σε μια γραμμή ροής λειτουργεί ένα μόνο αντίγραφο που προβλέπεται από τη διαδικασία κατασκευής και το κόστος ροής των υλικών είναι προς τον αριθμό των προϊόντων που παράγονται και των αποστάσεων. Επίσης η απόσταση μεταξύ διαδοχικών μηχανών είναι ίση μία μονάδα απόστασης. Σύμφωνα με τον Won[4] στο μοντέλο της Ελαχίστης Προς τα Πίσω Ροής επίσης ορίζονται τα παρακάτω:

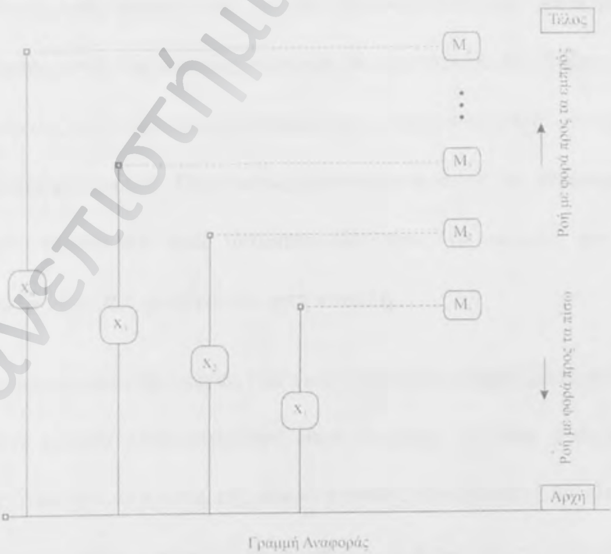
n : ο αριθμός των μηχανών σε μια κυψέλη παραγωγής.

m: ο αριθμός των προϊόντων εξαρτημάτων που κατασκευάζονται σε μια κυψέλη

d_j : η ζήτηση για το προϊόν j, $j = 1, 2, ..., m$

x_i : η απόσταση μεταξύ της Γραμμής Αναφοράς και της μηχανής i , $i = 1, 2, ..., n$

Το σχήμα 3 αναπαριστά μια τοποθέτηση μηχανών και του τρόπου υπολογισμού των σχετικών αποστάσεων ανάμεσα τους.



Σχήμα 3. Γραφική αναπαράσταση τοποθέτησης μηχανών.

[4] Won, You-Dong, A Linear Programming Approach to Linear Machine Layout Problem, The Journal of the Industrial Mathematics Society, 47, (1997), Part2.

Η απόσταση μεταξύ της Γραμμής Αναφοράς και της πρώτης μηχανής είναι ίση με 1 μονάδα απόστασης. Το μοντέλο με χρήση του γραμμικού προγραμματισμού εκφράζεται ως:

$$\min \sum_{h=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{m} d_j \left| x_h - x_i \right| \qquad \text{με} \qquad \begin{array}{l} h = 1, 2, \ldots, n \\ i = 1, 2, \ldots, n \\ j = 1, 2, \ldots, m \end{array} \tag{8.2}$$

Έτσι ώστε,

$$\left| x_h - x_i \right| \geq 1 \qquad \text{με} \qquad \begin{array}{l} h = 1, 2, \ldots, n \\ i = 1, 2, \ldots, n \\ h \neq i \end{array} \tag{8.3}$$

$$\text{και } x_i = 1, 2, \ldots, n \tag{8.4}$$

Η αντικειμενική συνάρτηση εκφράζεται με την εξίσωση (8.2) και απεικονίζει τη συνολική απόσταση μετακίνησης εντός της κυψέλης για όλα τα εξαρτήματα που πρέπει να παραχθούν. Η εξίσωση (8.3) εξασφαλίζει ότι δεν υπάρχει επικάλυψη μεταξύ δύο μηχανών και η εξίσωση (8.4) εισάγει τις μεταβλητές απόφασης. Στην αντικειμενική συνάρτηση το άθροισμα των συνολικών μετακινήσεων είναι το σύνολο των μετακινήσεων προς τα εμπρός και το σύνολο των μετακινήσεων προς τα πίσω που συμβαίνουν στη κυψέλη.

Το σύνολο των κινήσεων προς τα εμπρός (forward flow) δεν επηρεάζει τη ροή των υλικών και συνεπώς η ροή των υλικών είναι καλύτερη όταν οι προς τα πίσω ροές (backward flows) ελαχιστοποιούνται. Άρα μια έκφραση της αντικειμενικής συνάρτησης με ελαχιστοποίηση των προς τα πίσω ροών είναι καλύτερη και σε αυτό οφείλει το όνομα του το μοντέλο.

Το μοντέλο πρέπει να μετασχηματιστεί σε μια πιο κλασσική μορφή γραμμικού προγραμματισμού.

Έστω ο πίνακας Α και $A=[a_{hi}]$ και

$$a_{hi}=\begin{cases}1 & \text{εάν η μηχανή ι ακολουθεί τη μηχανη h}\\ 0 & \text{αλλιώς}\end{cases} \tag{8.5}$$

Επίσης έστω η f_{hi} ,η σταθμισμένη συχνότητα της ζήτησης d_j η οποία δίδεται από τη παρακάτω εξίσωση.

$$f_{hi}=\begin{cases}\sum_{i=1}^{n} a_{hij}\cdot d_j,\ h\neq i\\ 0,\ h=i\end{cases} \tag{8.6}$$

Συνεπώς η νέα μορφή της αντικειμενικής συνάρτησης είναι:

$$\min\sum_{h=1}^{n}\ \sum_{i=1}^{n} f_{hi}\left|x_h-x_i\right| \tag{8.7}$$

Η παραπάνω εξίσωση λαμβάνει υπόψη όλες τις κινήσεις εντός της κυψέλης. Αλλά επειδή μόνο οι προς τα πίσω κινήσεις είναι που έχουν αρνητικό αποτέλεσμα στη παραγωγή, η εξίσωση γίνεται :

$$\min\sum_{h=1}^{n}\sum_{i=1}^{n} f_{hi}\cdot\max\left|x_h-x_i,0\right| \tag{8.8}$$

έτσι ώστε :

$$\left|x_h-x_i\right|\geq 1 \qquad \text{με} \qquad \begin{matrix}h=1,2,...,n\\ i=h+1,h+2,...,n\end{matrix} \qquad \text{και} \qquad x_i=1,2,...n \tag{8.9}$$

Από το μοντέλο με τις απόλυτες τιμές και τον περιορισμό πρέπει να περάσουμε σε ένα ισοδύναμο 0-1 μοντέλο ακέραιου προγραμματισμού, ορίζοντας

$$z_{hi} = \max(x_h - x_i, 0) \tag{8.10}$$

Είναι προφανές ότι:

$$z_{hi} \geq 0 \tag{8.11}$$

$$z_{hi} \geq x_h - x_i \tag{8.12}$$

$$x_h - x_i \geq 1 - M\,\omega_{hi} \tag{8.13}$$

$$-x_h + x_i \geq M(1 - \omega_{hi}) \tag{8.14}$$

όπου Μ, ένας μεγάλος αριθμός και $\omega_{hi} = \begin{cases} 1 \\ 0 \end{cases}$

Τελικά, το 0-1 ακέραιου προγραμματισμού μοντέλου ελαχιστοποίησης της προς τα πίσω ροής είναι:

$$\min \sum_{h=1}^{n} \sum_{i=1}^{n} f_{hi} \cdot z_{hi} \tag{8.15}$$

έτσι ώστε,

$$x_h - x_i \geq 1 - M\,\omega_{hi} \qquad \text{με} \qquad \begin{matrix} h = 1, 2, \ldots, n \\ i = h+1, h+2, \ldots, n \\ h \neq i \end{matrix} \tag{8.16}$$

$$-x_h + x_i \geq M(1-\omega_{hi}) \quad \text{με} \quad \begin{array}{l} h = 1,2,\dots,n \\ i = h+1, h+2, \dots, n \\ h \neq i \end{array} \tag{8.17}$$

$$z_{hi} \geq x_h - x_i \quad \text{με} \quad \begin{array}{l} h = 1,2,\dots,n \\ i = h+1, h+2, \dots, n \end{array} \tag{8.18}$$

$$x_i = \sum_{j=1}^{n} j \cdot y_{ij} \quad \text{με} \quad i = 1,2,\dots,n \tag{8.19}$$

$$\sum_{j=1}^{n} y_{ij} = 1 \quad \text{με} \quad i = 1,2,\dots,n \tag{8.20}$$

$$x_i, z_{hi} \geq 0 \ \text{και} \ \omega_i, y_{hi} = \begin{cases} 0 \\ 1 \end{cases}$$

7.4.2 Εφαρμογή του Ant Colony System για το Πρόβλημα Τοποθέτησης Κατασκευαστικών Κυψελών

Από τον ορισμό του προβλήματος, δεν υπολογίζεται κόστος της απόσταση για τη ροή προς τα εμπρός. Σκοπός είναι η ελαχιστοποίηση της προς τα πίσω ροής με στην κυψέλη παραγωγής.

Απαιτούμενα Δεδομένα για την Ανάπτυξη Κυψέλης Παραγωγής

Η ανάπτυξη μιας κυψέλης παραγωγής περιλαμβάνει τρία βήματα.

- Δημιουργία μια οικογένειας εξαρτημάτων/προϊόντων που πρέπει να παραχθούν
- Τοποθέτηση μηχανών εντός της κυψέλης
- Τοποθέτηση κυψελών με βάση τις μεταξύ τους σχέσεις

Τα αρχικά δεδομένα που απαιτούνται είναι :

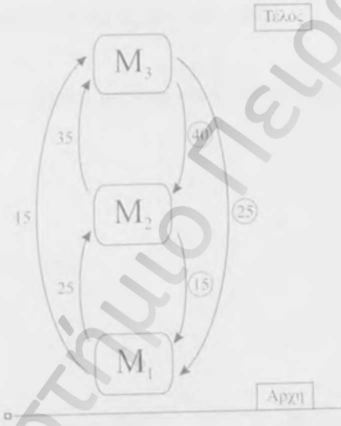
- Πίνακας Μηχανών/Εξαρτημάτων
- Ακολουθία των μηχανών που χρησιμοποιούνται για την παραγωγή κάθε εξαρτήματος

Για παράδειγμα όπως φαίνεται στο σχήμα 1, όταν μια διαδικασία παραγωγής καλεί τη χρήση της μηχανής 1 και μετά της μηχανής 2, ακόμα και αν μεταξύ τους άλλες μηχανές, το κόστος/απόσταση που θα λογιστεί στην αντικειμενική συνάρτηση είναι μηδέν. Αντίθετα, αν σε μια διεργασία απαιτείται να χρησιμοποιηθεί η μηχανή 3 και στη συνέχεια η μηχανή 2 το κόστος θα είναι ίσο με μια μονάδα απόστασης (καθώς $x_3 - x_2 = 1$) επί το τη ποσότητα του εξαρτήματος που κινείται. Με αυτό το τρόπο θα υπολογιστούν μια σειρά εσωτερικών αποστάσεων για μετακινήσεις εντός της κυψέλης, δηλαδή ένας πίνακας που θα έχει άμεση σχέση με την υλοποίηση του αλγορίθμου. Εάν έχουμε m μηχανές τότε θα έχουμε ένα πίνακα αποστάσεων $m \times m$ όπου κάθε στοιχείο του θα αναπαριστά το κόστος μετάβασης από τη μηχανή i στη μηχανή i+1. Από αυτό τον πίνακα τα μυρμήγκια θα επιλέξουν την επόμενη μηχανή που θα κινηθούν, δηλαδή που θα εντάξουν στη γραμμή ροής τους, και η διαδικασία θα συνεχιστεί μέχρι να επισκεφθεί όλες τις μηχανές που απαιτούνται για την παραγωγή του προϊόντος. Για τον υπολογισμό των στοιχείων αυτού του πίνακα, θεωρούμε δύο προϊόντων (P1, P 2) που για να παραχθούν χρησιμοποιούνται τρεις (M1,M2,M3) μηχανές. Ο παρακάτω πίνακας δείχνει την ακολουθία παραγωγής με βάση τους δείκτες των μηχανών καθώς και την ζήτηση για τα δύο προϊόντα

Προϊόν	Ακολουθία Παραγωγής	Ζήτηση
P 1	1 → 2 → 3 → 2 → 3 → 1	10 τεμάχια
P 2	3 → 2 → 1 → 3 – 2 → 3 → 1 → 2	15 τεμάχια

Πίνακας 3.Παράδειγμα υπολογισμού στοιχείων του πίνακα

Από το σχήμα 4 που απεικονίζει τη διάταξη και τις συνδέσεις των μηχανών και αναπαριστά το κόστος μετάβασης είτε για ροή προς τα εμπρός, είτε για ροή προς τα πίσω.



Σχήμα 4: Διάταξη και συνδέσεις των τριών μηχανών.

Οι πίνακες F και T αποτελούνται από τα στοιχεία για τις αποστάσεις μεταξύ των μηχανών για εμπρός και πίσω ροή. Από τον πίνακα F μπορούμε να υπολογίζουμε τις προς τα πίσω ροές λαμβάνοντας την ακολουθία παραγωγής από την τελευταία προς την πρώτη μηχανή. Ο πίνακας T είναι ο ανάστροφος του πίνακα F και από αυτόν υπολογίζουμε τις προς τα πίσω ροές άμεσα. Δηλαδή αν η παραγωγή ενός προϊόντος απαιτεί τη μετάβαση από τη μηχανή 1 στη μηχανή 3 και είναι τοποθετημένες διαδοχικά, τότε η προς τα πίσω ροή υπολογίζεται από τον πίνακα.

	M1	M2	M3
M1	0	25	15
M2	15	0	35
M3	25	40	0

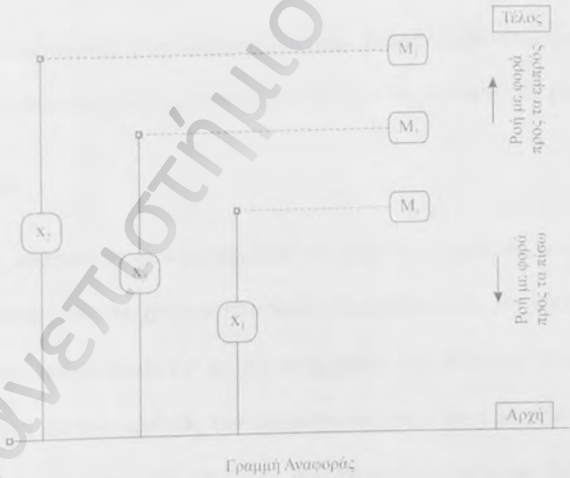
Πίνακας
F

	M1	M2	M3
M1	0	15	25
M2	25	0	40
M3	15	35	0

Πίνακας
T

Καθώς υπάρχει ο περιορισμός ότι δεν μπορεί να υπάρχει περισσότερο από μια φορά η ίδια μηχανή, ο πίνακας T δίνει απευθείας τις προς τα πίσω ροές.

Αν αποφασιστεί το σύνολο των τριών μηχανών να τοποθετηθούν σε μια γραμμή, όπως στο σχήμα 5 έχουμε τα παρακάτω.



Σχήμα 5: Ακολουθία μηχανών τοποθετημένων σε ευθεία γραμμή.

Ο παρακάτω πίνακας δείχνει τη διαδικασία που ακολουθείται για να παραχθεί το προϊόν 1. Με βάση τον ορισμό του προβλήματος προσμετράται κάθε οι κίνηση προς τα πίσω που για τη συγκεκριμένη διαδικασία στο σύνολο της είναι 3. Η ζήτηση για το προϊόν 1 είναι 10 τεμάχια και για κάθε ένα το κόστος από τη προς τα πίσω ροή είναι 3, συνεπώς το συνολικό κόστος για το εν λόγω προϊόν είναι 30. Από τη χωροδιάταξη για την πρώτη διεργασία που παράγει το προϊόν 1 προκύπτει ο πίνακας 4.

Διαδικασία Ροής	Υπολογισμός Αποστάσεων	Τιμή	Φορά Ροής
Γραμμή Αναφοράς στη M1	X1	1	Εμπρός
M1 στη M2	X2-X1	2	Εμπρός
M2 στη M3	X2-X3	1	Πίσω
M3 στη M2	X2-X3	1	Εμπρός
M2 στη M3	X2-X3	1	Πίσω
M3 στη M1	X3-X1	1	Πίσω
Σύνολο Ροών Προς τα Πίσω		3	

Πίνακας 4: Διαδικασία εύρεσης του κόστους και απόστασης για την πρώτη διεργασία

Για την εύρεση του συνολικού κόστους η διαδικασία επαναλαμβάνεται για κάθε προϊόν. Στη παρούσα υλοποίηση ο αριθμός των μηχανών είναι ίσος με τον αριθμό των μυρμηγκιών.

Ανάλυση Εφαρμογής

Με την έναρξη της εφαρμογής εισάγονται από το χρήστη ο αριθμός των προϊόντων που θα παραχθούν και μηχανών που θα χρησιμοποιηθούν. Ο αριθμός των μηχανών και των τεχνητών μυρμηγκιών είναι ίσος κατά αναλογία με τις εφαρμογές Ant Colony System σε προβλήματα περιοδεύοντος πωλητή όπου ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων. Κάθε μυρμήγκι επιλέγει με βάση ένα από τους δύο παρακάτω τρόπους. Επιλέγει μια διαδρομή με σαφή τρόπο, βάση της βέλτιστης απόδοσης αυτής που ορίζεται από την υψηλή ποσότητα φερομόνης και το χαμηλό κόστος, ή επιλέγει με βάση ένα πιθανοκρατικό κανόνα, μεταξύ των υποψηφίων διαδρομών. Η μαθηματική έκφραση αυτών των επιλογών δίδεται από τους κανόνες μετάβασης του αλγορίθμου που αναφέρθηκαν παραπάνω. Για παράδειγμα ,υποθέστε ότι πρέπει να παραχθούν πέντε προϊόντα P_1, P_2, P_3, P_4, P_5, με τρεις μηχανές M_1, M_2, M_3 την πρώτη φορά και έξι μηχανές $M_1, M_2, M_3, M_4, M_5, M_6$ τη δεύτερη φορά. Οι απαιτήσεις και η ακολουθία των διεργασιών για κάθε προϊόν δίνεται στο πίνακα 5.

Προϊόν	Ακολουθία Παραγωγής με 3 Μηχανές	Ακολουθία Παραγωγής με 6 Μηχανές	Τεμάχια
P_1	$1 \to 3 \to 2$	$1 \to 3 \to 4$	10
P_2	$2 \to 1 \to 2 \to 1 \to 2$	$2 \to 1 \to 2 \to 1 \to 2$	20
P_3	$3 \to 2 \to 1$	$3 \to 4 \to 1$	15
P_4	$3 \to 1 \to 2 \to 1$	$3 \to 4 \to 2 \to 1$	35
P_5	$3 \to 2 \to 1 \to 2 \to 1$	$3 \to 2 \to 4 \to 2 \to 1$	25

Πίνακας 5:Ακολουθιά παραγωγής για πέντε προϊόντα και ζήτηση.

Η ρύθμιση των περαμάτων του αλγορίθμου είναι ιδιαίτερα σημαντική για την απόδοση του, και ειδικά στον αλγόριθμο Ant Colony System όπου ο παράγοντας β είναι καθοριστικός για τη στάθμιση της ευριστικής πληροφορίας και στη παρούσα εφαρμογή είναι ίσος με 2. Επίσης, σημαντικός παράγοντας στην εφαρμογή των αλγορίθμων αποικίας μυρμηγκιών, είναι η επιλογή της κατάλληλης τιμής της αρχικής ποσότητας της φερομόνης τ_0 η οποία επηρεάζει συνήθως στα αρχικά στάδια του αλγορίθμου. Μια μικρή αρχική ποσότητα φερομόνης στις διαδρομές θα βοηθήσει στην εξερεύνηση διαδρομών και για αυτό ο παράγοντας τ_0 τίθεται ίσος με 0,1. Στον παρακάτω πίνακα δίνονται οι αποστάσεις μεταξύ των μηχανών που μπορεί να θεωρηθεί σαν ένας πίνακας κόστους.

$$\begin{array}{cc} & \text{To } i+1 \\ \text{From } i & \begin{bmatrix} 0 & 25 & 15 \\ 15 & 0 & 35 \\ 25 & 40 & 0 \end{bmatrix} \end{array} \quad F \qquad \begin{array}{cc} & \text{To } i \\ \text{From } i+1 & \begin{bmatrix} 0 & 15 & 25 \\ 25 & 0 & 40 \\ 15 & 35 & 0 \end{bmatrix} \end{array} \quad T$$

Στη συνέχεια υλοποιώντας το πρόγραμμα προκύπτουν τα αποτελέσματα της βέλτιστης ακολουθίας για κάθε προϊόν. Στους παρακάτω πίνακες παρουσιάζεται το καλύτερο κόστος για κάθε ροή προς τα πίσω και το μυρμήγκι – μηχανή που παράγει το καλύτερο κόστος.

Προϊόν	Κόστος	Μυρμήγκι Μηχανή	Ακολουθία Παραγωγής
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	75	2	2-1-3

Πίνακας 6: Αποτελέσματα ακολουθίας μηχανών για την παραγωγή με χρήση τριών μηχανών

Προϊόν	Κόστος	Μυρμήγκι Μηχανή	Ακολουθία Παραγωγής
1	60	1	1-3-4-6-5-2
2	80	2	2-4-1-3-6-5
3	90	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

Πίνακας 7. Αποτελέσματα ακολουθίας μηχανών για την παραγωγή με χρήση έξι μηχανών

Μετά το πέρας της εκτέλεσης του αλγορίθμου, η εφαρμογή έχει υπολογίσει το καλύτερο κόστος και δημιουργεί το ένα αρχείο όπου ανά επανάληψη και μυρμήγκι δείχνει το κόστος. Ο παρακάτω πίνακας δείχνει τη μορφή του αρχείου.

Iteration 1	
Ant 1 cost 140	
Machine Arrangement:	1 2 3
Ant 2 cost 105	
Machine Arrangement:	2 1 3
Ant 3 cost 70	
Machine Arrangement:	3 1 2

Πίνακας 8: Αποτελέσματα ακολουθίας μηχανών για την παραγωγή με χρήση τριών μηχανών στο τέλος της πρώτης επανάληψης.

Iteration 1						
Ant 1 cost 200						
Machine Arrangement:	1	3	4	6	5	2
Ant 2 cost 100						
Machine Arrangement:	2	4	1	3	6	5
Ant 3 cost 175						
Machine Arrangement:	3	4	1	6	5	2
Ant 4 cost 225						
Machine Arrangement:	4	1	3	6	5	2
Ant 5 cost 150						
Machine Arrangement:	5	1	3	4	6	2
Ant 6 cost 100						
Machine Arrangement:	6	1	3	4	2	5

Πίνακας 9: Αποτελέσματα ακολουθίας μηχανών για την παραγωγή με χρήση έξι μηχανών στο τέλος της πρώτης επανάληψης.

7.5 Συμπεράσματα

Η επιλογή των παραμέτρων του αλγορίθμου, όπως το αρχικό επίπεδο φερομόνης, είναι ιδιαίτερα σημαντική για τη σύγκλιση του αλγορίθμου καθώς παρατηρούνται σημαντικές διακυμάνσεις των αποτελεσμάτων τροποποιώντας ελάχιστα τις αρχικές ποσότητες φερομόνης. Στον αλγόριθμο Ant Colony System διαδικασία ανανέωσης της φερομόνης επηρεάζει και την σχετική απόδοση του αλγορίθμου. Στη συγκεκριμένη υλοποίηση, η διαδικασίας ανανέωσης φερομόνης, δίνει έμφαση στο ορισμό της αρχική ποσότητας φερομόνης τ_0. Είναι προφανές ότι η ποιότητα των αποτελεσμάτων είναι άμεσα συναρτώμενη των αρχικών τιμών της φερομόνης καθώς παρατηρούνται σημαντικές διακυμάνσεις των αποτελεσμάτων τροποποιώντας τις αρχικές ποσότητες φερομόνης. Δηλαδή με βάση τα δεδομένα του προβλήματος, η υψηλή τιμή 4000 δεν ενίσχυσε την εξερεύνηση (exploration) νέων λύσεων, παρά την ύπαρξη του συντελεστή εξάτμισης ΄ρ΄, αλλά πρότεινε την εκμετάλλευση (exploitation) παλαιών μη βέλτιστων λύσεων.

Αξιοσημείωτο είναι και το γεγονός ότι η αρχική τιμή φερομόνης ίση με 50 δεν επέτρεψε στους μηχανισμούς στιγμεργείας, μέσω θετικής ανάδρασης, να σχηματίσουν την βέλτιστη διαδρομή. Προφανώς για το συγκεκριμένο πρόβλημα και δεδομένα, ο ρυθμός εξάτμισης επικρατούσε πολύ γρήγορα και δεν επέτρεπε το σχηματισμό ίχνους φερομόνης που θα στρατολογούσε και άλλα μυρμήγκια. Τελικά, με αρχική τιμή φερομόνης 500 ελαχιστοποιούμε την αντικειμενική συνάρτηση του προβλήματος. Σημαντικό είναι το γεγονός ότι οι παράμετροι ‘β’ και ‘ρ’ παρέμειναν σταθεροί κατά την υλοποίηση με σκοπό η όλη συμπεριφορά του αλγορίθμου να εξαρτάται από την αρχική ποσότητα της φερομόνης η οποία ορίζει την ευελιξία του συστήματος στην εύρεση νέων, κάθε φορά, λύσεων.

Επίσης, παρατηρείται ότι μεγαλύτερος αριθμός επαναλήψεων του αλγορίθμου 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 γύρω από ένα σημείο. Στο σημείο αυτό, είναι πολύ δύσκολο (αν και όχι απίθανο) να βρεθεί μία σημαντικά καλύτερη λύση και λαμβάνεται η χαμηλότερη τιμή που έχει βρεθεί μέχρι τη στιγμή του τερματισμού ως η βέλτιστη.

Η εφαρμογή του μεταευριστικού πλαισίου ACO βασίζεται στο σωστό σχεδιασμό του αλγορίθμου και στη ρύθμιση των παραμέτρων σε σχέση με το υπό έρευνα πρόβλημα. Στη συγκεκριμένη εφαρμογή του Ant Colony System σκοπός είναι η βελτιστοποίηση των θαλάσσιων μεταφορών σε ένα σύνολο 13 κόμβων/πελατών με επιπρόσθετους περιορισμούς για τις απαιτήσεις των κόμβων.

Η ρύθμισή των παραμέτρων είναι κρίσιμη για την απόδοση του αλγορίθμου και η ανακάλυψη μιας περιοχής τιμών που δίνουν τα βέλτιστα βασίζεται σε πειραματικές επαναλήψεις της εφαρμογής. Η στάθμιση μεταξύ εξερεύνησης νέων λύσεων και εκμετάλλευσης παλαιοτέρων γίνεται από τις παραμέτρους β και q_0, που ρυθμίζουν το την κατασκευαστική συμπεριφορά του αλγορίθμου. Στην παρούσα υλοποίηση, η παράμετρος β πήρε την τιμή 1 και 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. Επίσης σημαντικό ρόλο έχει και ο παράγοντας 'ρ' της τεχνητής εξάτμισης της φερομόνης οποίος ήταν ίσος με 0.5. Σκοπός αυτού του παράγοντα είναι να εμποδίσει την υπερβολική συγκέντρωση φερομόνης στις διαδρομές, πράγμα που θα οδηγούσε σε πρόωρη σύγκλιση του αλγορίθμου σε μη βέλτιστες λύσεις.

Η υλοποίηση του αλγόριθμου Ant Colony System γίνεται έτσι ώστε να εφαρμοστεί στο πρόβλημα της τοποθέτησης μηχανών σε σειρά με σκοπό την εύρεση της βέλτιστης τοποθέτησης. Σημαντικά πλεονεκτήματα του μοντέλου της ελάχιστης προς τα πίσω ροής είναι η απλότητα του που κάνει εύκολο να υλοποιηθεί και το γεγονός ότι δεν έχει κόστος διότι χρησιμοποιεί δεδομένα που υπάρχουν στην παραγωγή. Όμως για την επιτυχή υλοποίηση τα δεδομένα που απαιτούνται είναι να υπάρχει σαφώς καθορισμένη διεργασία παραγωγής και σαφώς καθορισμένη ζητούμενη ποσότητα παραγωγής. Αλλαγές σε αυτούς τους παράγοντες οδηγεί σε αναδιοργάνωση της διάταξης των μηχανών. Ο τρόπος υλοποίησης και η αναφορά σε παραδοχές που σχετίζονται με το πρόβλημα της κυψελοειδούς παραγωγής δημιουργεί ένα πλαίσιο που με σχετικά λίγες αλλαγές μπορεί να εφαρμοστεί και σε άλλα προβλήματα κυψελοειδούς παραγωγής.

Πολύ σημαντικός παράγοντας στις υλοποιήσεις αλγορίθμων μυρμηγκιών είναι ο κατάλληλος προσδιορισμός των αρχικών παραμέτρων όπως η ποσότητα της φερομόνης η οποία μέσο της στιγμεργετικής επικοινωνίας θα οδηγήσει στην επιλογή των καλύτερων λύσεων έναντι των άλλων. Σε αυτή την αναζήτηση επίσης σημαντικό ρόλο παίζει και η ρύθμιση του παράγοντα 'β' στον κανόνα μετάβασης καθώς αυτός σταθμίζει τη φερομόνη και την ευριστική πληροφορία στη διαδικασία της επιλογής του επόμενου κόμβου-μηχανής. Η προσαρμοστικότητα του αλγορίθμου βασίζεται σε αυτούς τους παράγοντες. Η ισορροπία των παραμέτρων σε σχέση με το υπό έρευνα πρόβλημα σχετίζεται με την απόδοση του αλγορίθμου. Η εφαρμογή του Ant Colony System σε αυτά τα προβλήματα απέδωσε ικανοποιητικά αποτελέσματα αποδεικνύοντας την προσαρμοστικότητα και αποτελεσματικότητα του πλαισίου αλγορίθμων μυρμηγκιών καθώς τα υπολογιστικά αποτελέσματα δείχνουν ότι το πλαίσιο του αλγόριθμου Ant Colony System είναι μια αποτελεσματική μεταευριστική μέθοδος.

Κεφάλαιο 8

Μια Νέα Προσέγγιση του Αλγορίθμου Ant Colony System και Εφαρμογή για τη Βελτιστοποίηση του Προβλήματος του Περιοδεύοντος Πωλητή

8.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται μια νέα προσέγγιση του αλγόριθμου Ant Colony System και εφαρμογή του σε πρόβλημα περιοδεύοντος πωλητή. Το πρόβλημα του περιοδεύοντος πωλητή είναι ένα NP complete πρόβλημα όπου οι κόμβοι (πόλεις) του γραφήματος πρέπει να χρησιμοποιηθούν όλοι τουλάχιστον μια φορά στη δημιουργία της λύσης. Θεωρείται πρόβλημα αναφοράς και υπάρχουν έτοιμα στιγμιότυπα προβλημάτων (problem instances) για τη δοκιμή νέων ιδεών σε αυτό. Η πλέον γνωστή μορφοποίηση προβλήματος περιοδεύοντος πωλητή είναι η TSPLIB format, με την οποία δημιουργούνται τρία προβλήματα για 100, 500 και 1000 πόλεις της Ελλάδας. Προτείνεται ένας νέος αλγόριθμος ο οποίος δοκιμάζεται στα τρία προβλήματα με βάση ένα σύνολο αρχικών παραμέτρων. Στον νέο αλγόριθμο υπάρχουν καινούργιες διαδικασίες αλλά και διαδικασίες που εμπνέονται από παλαιότερους αλγορίθμους του πλαισίου ACO. Σαν συνέπεια αυτού η αρχική ρύθμιση των παραμέτρων γίνεται με βάση γνωστές τιμές που αποδίδουν πειραματικά και στη συνέχεια δοκιμάζονται νέες τιμές για τις παραμέτρους του αλγορίθμου με σκοπό την καλύτερη απόδοση στα τρία προβλήματα σε σχέση με τις λύσεις που δημιουργούνται από τον αλγόριθμο με τις αρχικές τιμές.

8.2 Έκφραση του Προβλήματος

Η κλασική περιγραφή του Προβλήματος του Περιοδεύοντος Πωλητή (Travelling Salesman Problem) είναι η παρακάτω: Ένας πωλητής πρέπει να επισκεφθεί n πόλεις, που απεικονίζονται με τους αριθμούς $1,2,\ldots,n$, ξεκινώντας από μια αρχική πόλη που θεωρούμε ως βάση, την πόλη 1, *αλλά* πρέπει να επισκεφθεί τις όλες υπόλοιπες $n-1$ πόλεις ακριβώς μία φορά και να επιστρέψει στη πόλη 1. Σκοπός είναι η δημιουργία μιας διαδρομής που ελαχιστοποιεί την απόσταση που χρειάζεται να ταξιδέψει ο πωλητής. Με όρους της θεωρίας γραφημάτων, αυτό το πρόβλημα είναι ανάλογο με την εύρεση ενός ελάχιστου Χαμιλτονιανού κυκλώματος (Hamiltonian circuit) σε πλήρες γράφημα $K_N \equiv G(N,A)$, όπου $N=\{1,2,\ldots,n\}$ είναι το σύνολο των κόμβων/κορυφών (nodes,vertices) και $A=\{(i,j): i,j \in N, i \neq j\}$ το σύνολο των τόξων που συνδέουν πλήρως τους κόμβους του συνόλου Ν. Ο πίνακας για το κόστος μετάβασης ή αποστάσεων θεωρείται θετικός, $C=[c_{ij}]$ όπου c_{ij} το κόστος μετάβασης ή απόσταση μεταξύ του κόμβου i και j. Η κατεύθυνση διάσχισης των κόμβων είναι σημαντική για το πρόβλημα. Όταν $c_{ij} \neq c_{ji}$ το πρόβλημα καλείται ασύμμετρο (asymmetric), ενώ όταν $c_{ij} = c_{ji}$ καλείται συμμετρικό (symmetric).

Στη συμμετρική εκδοχή του προβλήματος του περιοδεύοντος πωλητή (S-TSP) ισχύει ότι $c_{ij} = c_{ji}$ άρα η κατεύθυνση διάσχισης των κόμβων του γραφήματος δεν επηρεάζει το κόστος/μηκος της διαδρομής λύσης του προβλήματος. Συνεπώς θεωρούμε ένα μη κατευθυνόμενο γράφημα (undirected graph) όπου μόνο ένα τόξο συνδέει δύο κόμβους. Τότε ορίζουμε τη μεταβλητή απόφασης $x_j \in \{0,1\}$, όπου j αφορά όλες τις ακμές (edge) Ε του μη κατευθυνόμενου γραφήματος και c_j είναι το κόστος διάσχισης της ακμής j. Η εύρεση μιας διαδρομής λύση πρέπει να επιλεγεί από ένα υποσύνολο των ακμών έτσι ώστε κάθε κόμβος να περιέχεται σε ακριβώς δύο από τις ακμές που έχουμε επιλέξει. Οι παρακάτω συναρτήσεις εκφράζουν το συμμετρικό πρόβλημα.

$$\min \frac{1}{2} \sum_{j=1}^{n} \sum_{k \in J(j)} c_k x_k \tag{8.1}$$

με

$$\sum_{k \in J(j)} x_k = 2 \qquad \forall j = 1, 2, \dots, n \tag{8.2}$$

$$\sum_{j \in E(S)} x_j \le |S| - 1 \qquad \forall S \subset \{1, 2, \dots, n\} \tag{8.3}$$

$$x_j \in \{0, 1\} \qquad \forall j \in E(S)$$

όπου $J(j)$ το σύνολο των μη κατευθυνόμενων ακμών (undirected edges) που συνδέονται στο κόμβο j και $E(S)$ το υποσύνολο των μη κατευθυνόμενων ακμών που συνδέουν τις πόλεις που ανήκουν στο υποσύνολο των πόλεων S, με εφικτό τρόπο ώστε να μην δημιουργούνται υπογύροι.. Σε αυτή την έκφραση του προβλήματος υπάρχουν $n(n-1)/2$ δυαδικές μεταβλητές, δηλαδή οι μισές της ασύμμετρης περίπτωσης. Για να υπολογίσουμε τον πίνακα των αποστάσεων λαμβάνουμε την Ευκλείδεια Απόσταση, θεωρώντας ότι μία πόλη i έχει καρτεσιανές συντεταγμένες (Xi, Yi) και η επόμενή της j έχει συντεταγμένες (Xj, Yj), με βάση τον παρακάτω τύπο: $c_k = d_{ij} = \sqrt{(x_i - x_j)^2 (y_i - y_j)^2}$

Για την παρούσα υλοποίηση χρησιμοποιήθηκαν δεδομένα από το αρχείο gr9882.tsp[1] με μορφοποίηση TSPLIB EUC_2D για πρόβλημα με 100, 500 και 1000 πόλεις.

[1] http://www.tsp.gatech.edu/world/gr9882.tsp

```
NAME : gr100
COMMENT : 100 locations in Greece
COMMENT : Derived from National Imagery
and Mapping Agency data
TYPE : TSP
DIMENSION : 100
EDGE_WEIGHT_TYPE : EUC_2D
NODE_COORD_SECTION
1 34816.6667 24100.0000
2 34833.3333 24083.3333
3 34850.0000 24050.0000
4 34850.0000 24083.3333
5 34850.0000 24116.6667
6 34866.6667 24083.3333
7 34933.3333 24800.0000
8 34933.3333 24916.6667
9 34966.6667 24883.3333
10 34966.6667 24933.3333
11 34966.6667 24950.0000
12 34966.6667 25033.3333
13 34966.6667 25133.3333
14 34966.6667 25250.0000
15 34983.3333 24750.0000
16 34983.3333 24850.0000
17 34983.3333 24916.6667
18 34983.3333 24983.3333
19 34983.3333 25000.0000
20 34983.3333 25016.6667
21 34983.3333 25150.0000
22 34983.3333 25166.6667
23 34983.3333 25200.0000
24 34983.3333 25216.6667
25 34983.3333 25266.6667
26 34983.3333 25283.3333
27 34983.3333 25316.6667
28 34983.3333 25333.3333
29 35000.0000 24816.6667
30 35000.0000 24900.0000
31 35000.0000 24933.3333
32 35000.0000 25033.3333
33 35000.0000 25066.6667
34 35000.0000 25133.3333
35 35000.0000 25150.0000
36 35000.0000 25366.6667
37 35000.0000 25383.3333
38 35000.0000 25483.3333
39 35000.0000 25516.6667
40 35000.0000 25533.3333
41 35000.0000 25583.3333
42 35000.0000 25633.3333
43 35016.6667 24783.3333
44 35016.6667 24800.0000
45 35016.6667 24833.3333
46 35016.6667 24866.6667
47 35016.6667 24883.3333
48 35016.6667 24900.0000
49 35016.6667 24916.6667
50 35016.6667 24950.0000
51 35016.6667 24966.6667
52 35016.6667 25000.0000
53 35016.6667 25066.6667
54 35016.6667 25083.3333
55 35016.6667 25116.6667
56 35016.6667 25166.6667
57 35016.6667 25200.0000
58 35016.6667 25416.6667
59 35016.6667 25516.6667
60 35016.6667 25533.3333
61 35016.6667 25550.0000
62 35016.6667 25583.3333
63 35016.6667 25683.3333
64 35016.6667 25750.0000
65 35016.6667 25833.3333
66 35016.6667 26100.0000
67 35016.6667 26116.6667
68 35033.3333 24783.3333
69 35033.3333 24833.3333
70 35033.3333 24933.3333
71 35033.3333 25150.0000
72 35033.3333 25233.3333
73 35033.3333 25250.0000
74 35033.3333 25266.6667
75 35033.3333 25283.3333
76 35033.3333 25383.3333
77 35033.3333 25433.3333
78 35033.3333 25450.0000
79 35033.3333 25483.3333
80 35033.3333 25500.0000
81 35033.3333 25550.0000
82 35033.3333 25566.6667
83 35033.3333 25600.0000
84 35033.3333 25666.6667
85 35033.3333 25750.0000
86 35033.3333 25766.6667
87 35033.3333 25900.0000
88 35033.3333 25950.0000
89 35033.3333 25966.6667
90 35033.3333 25983.3333
91 35033.3333 26150.0000
92 35050.0000 24816.6667
93 35050.0000 24833.3333
94 35050.0000 24866.6667
95 35050.0000 24900.0000
96 35050.0000 24950.0000
97 35050.0000 25050.0000
98 35050.0000 25066.6667
99 35050.0000 25100.0000
100 35050.0000 25116.6667
EOF
```

Μορφοποίηση του προβλήματος για 100 πόλεις βάση του TSPLIB EUC_2D format.

8.3 Η Νέα Προσέγγιση του Αλγορίθμου Ant Colony System

Κάθε μυρμήγκι επιλέγει με βάση ένα από τους δύο παρακάτω τρόπους. Επιλέγει μια διαδρομή με σαφή τρόπο, βάση της βέλτιστης απόδοσης αυτής που ορίζεται από την υψηλή ποσότητα φερομόνης και το χαμηλό κόστος, ή επιλέγει με βάση ένα πιθανοκρατικό κανόνα, μεταξύ των υποψηφίων διαδρομών. Η μαθηματική έκφραση αυτών των επιλογών δίδεται παρακάτω.

Κατασκευή Λύσεων

Ο κανόνας μετάβασης για το μυρμήγκι k που βρίσκεται στη πόλη i και επιλέγει την πόλη j ορίζεται ως :

$$j = \begin{cases} \max_{l \in N_i^k} \{\tau_{il}(t) \cdot [\eta_{il}(t)]^{\beta}\}, & \text{αν } q \le q_0 \\ J & \text{, αν } q > q_0 \end{cases} \qquad (8.4)$$

Όπου : q μια τυχαία μεταβλητή ομοιόμορφα κατανεμημένη στο [0,1]

q_0 παράμετρος που ορίζεται από τον ερευνητή στο [0,1], $0 \le q_0 \le 1$

J κόμβος που επιλέγεται με βάση τον πιθανοκρατικό κανόνα επιλογής αλλά με α=1

Ο πιθανοκρατικός κανόνας επιλογής ορίζεται ως:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}(t)]^{\beta}} & \text{, εάν } j \in N_i^k \\ 0 & \text{, εάν } j \notin N_i^k \end{cases} \qquad (8.5)$$

Όπου: τ_{ij} η ποσότητα της φερομόνης στην ακμή (i, j)

η_{ij} η ποσότητα της ευριστικής πληροφορίας

α, β παράμετροι που σταθμίζουν ευριστική πληροφορία και φερομόνη

N_i^k οι πόλεις που μπορεί να επισκεφθεί το μυρμήγκι

Στη συγκεκριμένη υλοποίηση χρησιμοποιούμε $\eta_{ij} = \dfrac{1}{d_{ij}}$

Όταν $q \leq q_0$, επιλέγεται η εκμετάλλευση της είδη υπάρχουσας λύσης-γνώσης στον εξερευνημένο χώρο, που έχει δημιουργηθεί από τη συλλογική συσσώρευση φερομόνης στις διαδρομές. Δηλαδή επιλέγουν τοπικά βέλτιστες λύσεις με συνέπεια να μην ευρεθεί μια νέα πιθανά ολικά βέλτιστη λύση.

Όταν $q > q_0$, επιλέγεται η εξερεύνηση του χώρου αναζήτησης και γίνεται προσπάθεια να επιλεγεί η καλύτερη τη συγκεκριμένη χρονική στιγμή λύση με βάση μια πιθανότητα. Επίσης, η μέθοδος θεωρεί πιθανό το ενδεχόμενο κάποιο μυρμήγκι να αποκλίνει σημαντικά (δηλαδή να 'χαθεί') και να ανακαλύψει μια νέα καλύτερη διαδρομή.

Με πιθανότητα q_0 επιλέγεται ο καλύτερος επόμενος κόμβος με βάση την 'μνήμη' του αλγορίθμου που αποτυπώνεται στα ίχνη φερομόνης. Ο ορισμός του q_0 είναι καθοριστικός για την εξερευνητική συμπεριφορά του αλγορίθμου, καθώς επιλέγουμε αν θα εστιάσει σε υπάρχουσες βέλτιστες διαδρομές ή θα εξερευνήσει νέες διαδρομές. Είναι προφανές ότι ο ορισμός της παραμέτρου q_0 είναι ιδιαίτερα σημαντικός για την καλή απόδοση του αλγορίθμου.

Ανανέωση της Φερομόνης

Στον κλασσικό Ant Colony System η εναπόθεση φερομόνης επιτρέπεται μόνο από το μυρμήγκι που έχει ανακαλύψει την καλύτερη διαδρομή μέχρι εκείνη τη χρονική στιγμή. Η διαδικασία αυτή έχει ως σκοπό να προκαλέσει συμπεριφορά εξερεύνησης στον αλγόριθμο με στόχο νέα ολικά βέλτιστη λύση. Στη συγκεκριμένη υλοποίηση προτείνεται μια νέα διαφοροποιημένη εξίσωση ανανέωσης της φερομόνης που δίδεται παρακάτω

Στη νέα προσέγγιση του αλγορίθμου ACS, η εξίσωση που εκφράζει την ανανέωση της φερομόνης σε κάθε ακμή (i, j) μετά τη διάσχισή του είναι:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \tau_0 \tag{8.6}$$

όπου:

τ_{ij} είναι η ποσότητα φερομόνης σε κάθε ακμή (i, j)

τ_0 είναι η αρχική ποσότητα φερομόνης

ρ είναι ο συντελεστής εξάτμισης φερομόνης με : $0 \le \rho \le 1$

Παράμετροι

Με τη νέα μορφή της διαδικασίας ανανέωσης φερομόνης, ιδιαίτερο ρόλο λαμβάνει η σχέση της με τον κανόνα μετάβασης καθώς το επίπεδο της φερομόνης σε μία ακμή (i, j) έχει βαρύτητα στη διαδικασία επιλογής του επόμενου κόμβου που ρυθμίζεται από το συντελεστή α ,ο οποίος όπως και στον παλαιό ACS έχει τιμή ίση με 1. Ο κανόνας μετάβασης νέου αλγορίθμου δίνει μεγαλύτερο βάρος στην ευριστική πληροφορία που ρυθμίζεται από το συντελεστή β .

Σχετικά υψηλά επίπεδα φερομόνης μπορεί να εγκλωβίσουν τον αλγόριθμο σε τοπικά βέλτιστα ενώ σχετικά χαμηλά να οδηγήσουν σε μεγάλο υπολογιστικό χρόνο. Ο ρυθμός εξάτμισής ρ έχει σαν σκοπό να αποτρέψει μια τέτοια εξέλιξη και είναι ένας ρυθμιστικός παράγοντας με σημαντικό ρόλο στη διαφοροποιημένη διαδικασία ανανέωσης της φερομόνης. Όπως θα δούμε παρακάτω διαφορετικά επίπεδα εξάτμισης ρ επηρεάζουν την απόδοση του αλγορίθμου.

Επιπρόσθετα, σημαντικό είναι το μέγεθος του προβλήματος σε σχέση με το μέγεθος της αποικίας που χρησιμοποιείται. Δηλαδή, ο αριθμός των πόλεων n σε σχέση με τον αριθμό των μυρμηγκιών m. Σαν βέλτιστές πειραματικές τιμές έχουν προταθεί[2] για τον Ant Colony System ένας αριθμός πολύ μικρότερος του προβλήματος, τις περισσότερες φορές ίσος με 10 ,αλλά στον Ant System τίθεται ίσος ο αριθμός των μυρμηγκιών με των αριθμό των πόλεων.

Ένας αλγόριθμος της οικογένειας Ant Colony Optimization είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν. Η ρύθμιση αυτού του μηχανισμού εξαρτάται από τα διαφορετικά τμήματα που τον απαρτίζουν καθώς και τις παραμέτρους που έχει το κάθε τμήμα. Η μακροσκοπική συμπεριφορά του αλγορίθμου για παράδειγμα, εξαρτάται είτε από τον τρόπο που τα μυρμήγκια κατασκευάζουν τις λύσεις τους με βάση τον πιθανοκρατικό κανόνα μετάβασης, είτε από την διαδικασία ανανέωσης της φερομόνης πάντα σε σχέση με το υπό έρευνα πρόβλημα. Επειδή λοιπόν αποτελούνται από σύνθετα τμήματα που αλληλεπιδρούν είναι δύσκολο να προβλέψουμε την συμπεριφορά τους σε καινοτόμα προβλήματα.

[2] Dorigo, M., Stützle, T., (2004). Ant Colony Optimization, MIT Press, Cambridge, Massachusetts

8.4 Εφαρμογή της Νέας Προσέγγισης του Ant Colony System για τη Βελτιστοποίηση του Προβλήματος του Περιοδεύοντος Πωλητή

Η υλοποίηση βασίστηκε σε δημόσιο ανοικτό λογισμικό[3], με σκοπό την βέλτιστη κωδικοποίηση των διαδικασιών της νέας μορφής του Ant Colony System.

Παρακάτω δίδεται ο ψευδοκώδικας για τη διαδικασία κατασκευής λύσεων.

```
Procedure Construct Solutions  New ACS

    for k=1 to m do
        for i=1 to k n do
            ant[k].visited[i]←false
        end-for
    end-for
    step ← 1
    for k=1 to m do
        r ← random{1,...,n}
        ant[k].tour[step] ← r
        ant[k].visited[r] ← true
    end-for
    while (step<n) do
        step ← step + 1
```

[3] http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html

```
            for k=1 to m do
                    ACSDecisionRule(k,step)
            end-for
        end-while
        for k=1 to m do
            ant[k].tour[n+1] ← ant[k].tour[1]
            ant[k].tour_length ← ComputeTourLength
        end-for
end-procedure
```

Παρακάτω δίδεται ο ψευδοκώδικας για τη νέα διαδικασία ανανέωσης της φερομόνης.

```
Procedure Update Pheromones New ACS

        for i=1 to n do

            for j=1 to n do

                    pheromone[i][j] ← (1-ρ) · pheromone[i][j] + τ0
                    pheromone[j][i] ← pheromone[i][j]

            end-for

        end-for
end procedure
```

Αρχικές Τιμές Παραμέτρων

Ο νέος αλγόριθμος έχει καινούργιες διαδικασίες αλλά και διαδικασίες από τον Ant Colony System. Η διαδικασία κατασκευής λύσεων είναι ίδια με του Ant Colony System και συνεπώς σαν εναρκτήριο σημείο πειραματισμού με το νέο αλγόριθμο οι τιμές των παραμέτρων θα τεθούν με τις πειραματικά προτεινόμενες του αρχικού αλγορίθμου. Η διαδικασία ανανέωσης της φερομόνης αν και αντλεί έμπνευση από υπάρχοντες υλοποιήσεις του πλαισίου ACO, με κοντινότερη την υλοποίηση Ant System, είναι εντελώς καινούργια. Συνεπώς, πλήθος τιμών μπορούν να δοκιμαστούν κατά την εφαρμογή του αλγορίθμου αν και σαν αρχικές τιμές για τις παραμέτρους της διαδικασίας θα χρησιμοποιηθούν τιμές κοντά στις πειραματικά προτεινόμενες του Ant System. Στην υλοποίηση του νέου αλγορίθμου δε συμπεριλαμβάνεται μέθοδος τοπικής έρευνας.

Οι παράμετροι β και q_0 επηρεάζουν τη διαδικασία κατασκευής λύσεων του νέου αλγορίθμου και οι βέλτιστες προτεινόμενες τιμές για αυτές είναι μεταξύ 2 και 5 για τη παράμετρο β και 0.9 για τη παράμετρο q_0.

Για τη νέα διαδικασία ανανέωσης της φερομόνης ο παράγοντας εξάτμισης ρ έχει σαν σκοπό να βοηθά την έρευνα για νέες λύσεις απαλείφοντας από την μνήμη της αποικίας, που είναι ο πίνακας φερομόνης τις λιγότερο ποιοτικές λύσεις. Οι αρχικές τιμές της παραμέτρου λαμβάνονται από προτεινόμενες τιμές για υλοποιήσεις του Ant System και του Ant Colony System είναι αντίστοιχα 0,5 και 0,1. Εννοιολογικά η νέα διαδικασία ανανέωσης της φερομόνης είναι πιο κοντά στον Ant System από ότι στον Ant Colony System και συνεπώς επιλέγεται σαν αρχική τιμή η 0,5.

8.5 Αποτελέσματα της Νέας Προσέγγισης του ACS για τη Βελτιστοποίηση του Προβλήματος του Περιοδεύοντος Πωλητή

8.5.1 Αποτελέσματα με τις Αρχικές Τιμές Παραμέτρων

Τα αποτελέσματα για κάθε πρόβλημα αποτελούν ένα εφαλτήριο για τις επόμενες δοκιμές του αλγορίθμου με νέες παραμέτρους ανά στιγμιότυπο του προβλήματος. Οι παράμετροι που δοκιμάζονται επηρεάζουν διαφορετικά τμήματα της λειτουργίας του αλγορίθμου, όπως για παράδειγμα οι παράμετροι οι β και q_0 που είναι ανήκουν στη διαδικασία κατασκευής λύσεων ή η παράμετρος ρ που επηρεάζει τη διαδικασία ανανέωσης της φερομόνης. Για τα στιγμιότυπα του προβλήματος με 100, 500 και 100 πόλεις δοκιμάστηκε για δέκα δοκιμές (tries) η νέα προσέγγιση του αλγορίθμου με τις αρχικές παραμέτρους που αναφέρονται παραπάνω. Για κάθε δοκιμή παρέχονται στατιστικές πληροφορίες για την κατασκευή της καλύτερης έως τώρα λύσης (best-so-far solution) και κάθε φορά που βρίσκει μια νέα καλύτερη λύση δίνει τα αποτέλεσμα με τη μορφή :

best <number> iteration <number> tours <number> time <number>

όπου :

best είναι το μήκος της διαδρομής (tour) της καλύτερης έως τώρα λύσης iteration είναι ο αριθμός της επανάληψης που βρέθηκε η καλύτερη λύση

tours είναι ο αριθμός των λύσεων που έχουν κατασκευαστεί (iteration X n_ants)

time είναι ο χρόνος που βρέθηκε η καλύτερη λύση

Ο αλγόριθμος τερματίζει όταν όλα τα μυρμήγκια κατασκευάσουν μια λύση ή μετά από 10 δευτερόλεπτα. Παρακάτω παρουσιάζονται τα αποτελέσματα τις Αρχικές Τιμές Παραμέτρων.

Στη πλειοψηφία των υλοποιήσεων του πλαισίου ACO ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων. Είναι δηλαδή $m = n$, με την εξαίρεση του Ant Colony System όπου προτείνεται $m = 10$. Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι $1/10$, $1/2$, $1/1$ αλλά σαν αρχική επιλέγεται η $m = n$.

Στο παρακάτω πίνακα δίδονται οι αρχικές τιμές των παραμέτρων που δοκιμάστηκαν και στα τρία στιγμιότυπα του προβλήματος.

Παράμετρος	**Αρχική Τιμή Παραμέτρου**
β	2
q_0	0,9
ρ	0,1
m	n

Πίνακας 1

Είναι σαφές ότι σε κάθε στιγμιότυπο του προβλήματος διαφορετικές τιμές παραμέτρων θα αποδώσουν καλύτερα ή χειρότερα ανάλογα με το πρόβλημα.

8.5 Αποτελέσματα της Νέας Προσέγγισης του ACS για τη Βελτιστοποίηση του Προβλήματος του Περιοδεύοντος Πωλητή

8.5.1 Αποτελέσματα με τις Αρχικές Τιμές Παραμέτρων

Τα αποτελέσματα για κάθε πρόβλημα αποτελούν ένα εφαλτήριο για τις επόμενες δοκιμές του αλγορίθμου με νέες παραμέτρους ανά στιγμιότυπο του προβλήματος. Οι παράμετροι που δοκιμάζονται επηρεάζουν διαφορετικά τμήματα της λειτουργίας του αλγορίθμου, όπως για παράδειγμα οι παράμετροι οι β και q_0 που είναι ανήκουν στη διαδικασία κατασκευής λύσεων ή η παράμετρος ρ που επηρεάζει τη διαδικασία ανανέωσης της φερομόνης. Για τα στιγμιότυπα του προβλήματος με 100, 500 και 100 πόλεις δοκιμάστηκε για δέκα δοκιμές (tries) η νέα προσέγγιση του αλγορίθμου με τις αρχικές παραμέτρους που αναφέρονται παραπάνω. Για κάθε δοκιμή παρέχονται στατιστικές πληροφορίες για την κατασκευή της καλύτερης έως τώρα λύσης (`best-so-far solution`) και κάθε φορά που βρίσκει μια νέα καλύτερη λύση δίνει τα αποτέλεσμα με τη μορφή :

```
best <number>      iteration <number>      tours <number>    time <number>
```

όπου :

`best` είναι το μήκος της διαδρομής (`tour`) της καλύτερης έως τώρα λύσης `iteration` είναι ο αριθμός της επανάληψης που βρέθηκε η καλύτερη λύση

`tours` είναι ο αριθμός των λύσεων που έχουν κατασκευαστεί (`iteration X n_ants`)

`time` είναι ο χρόνος που βρέθηκε η καλύτερη λύση

Ο αλγόριθμος τερματίζει όταν όλα τα μυρμήγκια κατασκευάσουν μια λύση ή μετά από 10 δευτερόλεπτα. Παρακάτω παρουσιάζονται τα αποτελέσματα τις Αρχικές Τιμές Παραμέτρων.

Try	Best	Iteration	Time	Tot.Time
1	Best: 5252	Iterations: 3321	Time 7.46	Tot.time 10.00
2	Best: 5257	Iterations: 2026	Time 4.56	Tot.time 10.00
3	Best: 5252	Iterations: 4043	Time 9.14	Tot.time 10.00
4	Best: 5244	Iterations: 1961	Time 4.43	Tot.time 10.00
5	Best: 5191	Iterations: 213	Time 0.48	Tot.time 10.00
6	**Best: 5174**	**Iterations: 2607**	**Time 5.89**	**Tot.time 10.00**
7	Best: 5238	Iterations: 685	Time 1.54	Tot.time 10.00
8	Best: 5227	Iterations: 83	Time 0.19	Tot.time 10.00
9	Best: 5200	Iterations: 2203	Time 5.03	Tot.time 10.00
10	Best: 5188	Iterations: 419	Time 0.94	Tot.time 10.00

Πίνακας 2. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με τις Αρχικές Τιμές Παραμέτρων.

Try	Best	Iteration	Time	Tot.Time
1	Best: 16108	Iterations: 2	Time 0.18	Tot.time 10.06
2	Best: 16138	Iterations: 105	Time 9.41	Tot.time 10.07
3	Best: 15947	Iterations: 7	Time 0.61	Tot.time 10.03
4	**Best: 15844**	**Iterations: 1**	**Time 0.09**	**Tot.time 10.06**
5	Best: 15951	Iterations: 3	Time 0.26	Tot.time 10.08
6	Best: 16043	Iterations: 8	Time 0.69	Tot.time 10.01
7	Best: 16098	Iterations: 1	Time 0.09	Tot.time 10.08
8	Best: 16105	Iterations: 2	Time 0.17	Tot.time 10.03
9	Best: 16252	Iterations: 38	Time 3.26	Tot.time 10.07

Πίνακας 3. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με τις Αρχικές Τιμές Παραμέτρων.

Try	Best	Iteration	Time	Tot.Time
1	Best: 30508	Iterations: 6	Time 3.59	Tot.time 10.02
2	Best: 30258	Iterations: 1	Time 0.59	Tot.time 10.59
3	Best: 30593	Iterations: 2	Time 1.25	Tot.time 10.04
4	**Best: 30188**	**Iterations: 6**	**Time 3.90**	**Tot.time 10.08**
5	Best: 30360	Iterations: 1	Time 0.57	Tot.time 10.02
6	Best: 30347	Iterations: 1	Time 0.64	Tot.time 10.26
7	Best: 30447	Iterations: 6	Time 3.55	Tot.time 10.08
8	Best: 30600	Iterations: 1	Time 0.61	Tot.time 10.09
9	Best: 30613	Iterations: 7	Time 4.17	Tot.time 10.17

Πίνακας 4. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με τις Αρχικές Τιμές Παραμέτρων.

Παρακάτω παρουσιάζονται τα αποτελέσματα με νέες τιμές για τις παραμέτρους β , q_0 , ρ και m ανά στιγμιότυπο του προβλήματος.

8.5.2 Αποτελέσματα για το Πρόβλημα με 100 Πόλεις

Παράμετρος β

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου β ίσες με 3,5 και 5.

Try	Best	Iteration	Time	Tot.Time
2	**Best: 5127**	**Iterations: 490**	**Time 1.52**	**Tot.time 10.00**
9	Best: 5164	Iterations: 2330	Time 7.30	Tot.time 10.00
10	Best: 5173	Iterations: 2142	Time 6.66	Tot.time 10.00

Πίνακας 5. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με β ίσο με 3.5.

Try	Best	Iteration	Time	Tot.Time
8	**Best: 5173**	**Iterations: 2179**	**Time 6.79**	**Tot.time 10.00**
3	Best: 5175	Iterations: 2505	Time 7.78	Tot.time 10.00
7	Best: 5176	Iterations: 1596	Time 4.95	Tot.time 10.00

Πίνακας 6. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με β ίσο με 5.

Οι προτεινόμενες καλύτερες τιμές για τον παράγοντα είναι μεταξύ 2 και 5. Για το πρόβλημα με τις 100 πόλεις με τις αρχικές τιμές των παραμέτρων, δηλαδή για β ίσο με 2, η καλύτερη λύση που βρέθηκε έχει τιμή 5174. Συνεπώς, σε σχέση με τις αρχικές παραμέτρους, αυξάνοντας την επιρροή της ευριστικής πληροφορίας θα έχουμε καλύτερα αποτελέσματα καθώς και οι δύο λύσεις με μεγαλύτερες τιμές του παράγοντα β είναι καλύτερες.

Παράμετρος q_0

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου q_0 ίσες με 0,3 και 0,6.

Try	Best	Iteration	Time	Tot.Time
8	**Best: 5601**	**Iterations: 1**	**Time 0.00**	**Tot.time 10.00**
2	Best: 5710	Iterations: 2107	Time 5.72	Tot.time 10.00
5	Best: 5713	Iterations: 1	Time 0.00	Tot.time 10.00

Πίνακας 7. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με q_0 ίσο με 0,3.

Try	Best	Iteration	Time	Tot.Time
5	**Best: 5286**	**Iterations: 3786**	**Time 9.35**	**Tot.time 10.00**
8	Best: 5318	Iterations: 1	Time 0.00	Tot.time 10.00
9	Best: 5359	Iterations: 1555	Time 3.83	Tot.time 10.00

Πίνακας 8. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με q_0 ίσο με 0,6.

Η αρχική τιμή του q_0 είναι ίση με 0,9 και προτείνεται από τον απλό ACS. Προφανώς πρόκειται για τιμή που δίνει καλά αποτελέσματα καθώς οι λύσεις που δημιουργήθηκαν είναι κατώτερες και για τις δύο εναλλακτικές τιμές της παραμέτρου.

Παράμετρος ρ

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμή της παραμέτρου ρ ίση με 0,5.

Try	Best	Iteration	Time	Tot.Time
6	**Best: 5238**	**Iterations: 2**	**Time 0.00**	**Tot.time 10.00**
5	Best: 5255	Iterations: 4	Time 0.01	Tot.time 10.00
8	Best: 5268	Iterations: 253	Time 0.55	Tot.time 10.00

Πίνακας 9. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με ρ ίσο με 0,5.

Από τα αποτελέσματα, συμπεραίνουμε ότι υψηλότερος ρυθμός εξάτμισης δεν οδηγεί σε καλύτερο από το αρχικό αποτέλεσμα για το πρόβλημα με 100 πόλεις.

Παράμετρος *m*

Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι $1/10$, $1/2$, $1/1$ αλλά σαν αρχική επιλέγεται η $m = n$. Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου m ίσες με 10 και 50.

Try	Best	Iteration	Time	Tot.Time
10	**Best: 5168**	**Iterations: 20999**	**Time 4.61**	**Tot.time 10.00**
4	Best: 5169	Iterations: 26890	Time 5.86	Tot.time 10.00
3	Best: 5187	Iterations: 21923	Time 4.76	Tot.time 10.00

Πίνακας 10. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με m ίσο με 10.

Try	Best	Iteration	Time	Tot.Time
8	**Best: 5197**	**Iterations: 6542**	**Time 7.18**	**Tot.time 10.00**
2	Best: 5202	Iterations: 5502	Time 6.07	Tot.time 10.00
6	Best: 5206	Iterations: 4611	Time 5.08	Tot.time 10.00

Πίνακας 11. Αποτελέσματα για το πρόβλημα με 100 Πόλεις με m ίσο με 50.

Θέτοντας την παράμετρο m ίση με δέκα, δηλαδή ο αριθμός των μυρμηγκιών να είναι το $1/10$ των πόλεων, δημιουργείται μια λύση ανώτερη από την αρχική που έχει τιμή 5174. Όμως θέτοντας την παράμετρο ίση με 50 δεν έχει το ίδιο αποτέλεσμα καθώς η λύση που δημιουργείται είναι οριακά χειρότερη της αρχικής.

8.5.3 Αποτελέσματα για το Πρόβλημα με 500 Πόλεις

Παράμετρος β

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου β ίσες με 3,5 και 5.

Try	Best	Iteration	Time	Tot.Time
3	**Best: 15567**	**Iterations: 16**	**Time 1.76**	**Tot.time 10.04**
1	Best: 15701	Iterations: 10	Time 1.12	Tot.time 10.07
6	Best: 15703	Iterations: 2	Time 0.24	Tot.time 10.06

Πίνακας 12. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με β ίσο με 3,5.

Try	Best	Iteration	Time	Tot.Time
4	**Best: 15530**	**Iterations: 2**	**Time 0.23**	**Tot.time 10.10**
1	Best: 15598	Iterations: 80	Time 8.89	Tot.time 10.07
7	Best: 15601	Iterations: 6	Time 0.68	Tot.time 10.08

Πίνακας 13. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με β ίσο με 5.

Και οι δύο τιμές που δοκιμάζονται δημιουργούν καλύτερη λύση από την αρχική που έχει τιμή 15844. Για το πρόβλημα με 500 πόλεις και οι τρείς καλύτερες λύσεις που δημιουργούνται για β ίσο με 3,5 *αλλά* και 5, είναι καλύτερες της αρχικής.

Παράμετρος q_0

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου q_0 ίσες με 0,3 και 0,6.

Try	Best	Iteration	Time	Tot.Time
9	**Best: 18574**	**Iterations: 1**	**Time 0.10**	**Tot.time 10.03**
2	Best: 18579	Iterations: 1	Time 0.13	Tot.time 10.08
8	Best: 18684	Iterations: 1	Time 0.10	Tot.time 10.01

Πίνακας 14. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με q_0 ίσο με 0,3.

Try	Best	Iteration	Time	Tot.Time
5	**Best: 17066**	**Iterations: 1**	**Time 0.11**	**Tot.time 10.17**
2	Best: 17207	Iterations: 1	Time 0.10	Tot.time 10.25
4	Best: 17261	Iterations: 1	Time 0.10	Tot.time 10.12

Πίνακας 15. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με q_0 ίσο με 0,6.

Η αρχική τιμή του q_0 είναι ίση με 0,9 και προτείνεται από τον απλό ACS, δίνει καλύτερα αποτελέσματα.

Παράμετρος ρ

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμή της παραμέτρου ρ ίση με 0,5.

Try	Best	Iteration	Time	Tot.Time
9	**Best: 15814**	**Iterations: 69**	**Time 5.99**	**Tot.time 10.00**
10	Best: 15945	Iterations: 13	Time 1.12	Tot.time 10.07
8	Best: 15992	Iterations: 2	Time 0.18	Tot.time 10.09

Πίνακας 16. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με ρ ίσο με 0,5.

Για το πρόβλημα των 500 πόλεων η νέα τιμή της παραμέτρου οδηγεί σε οριακά καλύτερη λύση από την αρχική.

Παράμετρος m

Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι $1/10$, $1/2$, $1/1$ αλλά σαν αρχική επιλέγεται η $m=n$. Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου m ίσες με 50 και 250.

Try	Best	Iteration	Time	Tot.Time
5	**Best: 15702**	**Iterations: 2**	**Time 0.02**	**Tot.time 10.00**
7	Best: 15902	Iterations: 47	Time 0.38	Tot.time 10.00
1	Best: 15948	Iterations: 10	Time 0.09	Tot.time 10.00

Πίνακας 17. Αποτελέσματα για το πρόβλημα με 500 Πόλεις με m ίσο με 50.

Try	Best	Iteration	Time	Tot.Time
10	**Best: 15737**	**Iterations: 1**	**Time 0.05**	**Tot.time 10.02**
6	Best: 15844	Iterations: 2	Time 0.09	Tot.time 10.00
5	Best: 15901	Iterations: 7	Time 0.30	Tot.time 10.03

Πίνακας 18 Αποτελέσματα για το πρόβλημα με 500 Πόλεις με m ίσο με 250.

Για το πρόβλημα με τις 500 πόλεις ,σε αντίθεση με το προηγούμενο στιγμιότυπο του προβλήματος, και οι δύο τιμές που δοκιμάζονται δημιουργούν καλύτερη λύση από την αρχική.

8.5.4 Αποτελέσματα για το Πρόβλημα με 1000 Πόλεις

Παράμετρος β

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου β ίσες με 3,5 και 5.

Try	Best	Iteration	Time	Tot.Time
9	**Best: 29469**	**Iterations: 4**	**Time 2.91**	**Tot.time 10.41**
8	Best: 29699	Iterations: 2	Time 1.48	Tot.time 10.42
7	Best: 29838	Iterations: 4	Time 2.84	Tot.time 10.40

Πίνακας 19. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με β ίσο με 3,5.

Try	Best	Iteration	Time	Tot.Time
5	**Best: 29328**	**Iterations: 2**	**Time 1.39**	**Tot.time 10.01**
6	Best: 29575	Iterations: 9	Time 6.17	Tot.time 10.34
1	Best: 29584	Iterations: 7	Time 4.72	Tot.time 10.30

Πίνακας 20. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με β ίσο με 5.

Για το πρόβλημα με 1000 πόλεις οι τρείς καλύτερες λύσεις που δημιουργούνται για β ίσο με 3,5 αλλά και 5, είναι καλύτερες της αρχικής που έχει τιμή 30188. Συνεπώς, περισσότερη επιρροή της ευριστικής πληροφορίας οδηγεί τον αλγόριθμο σε σημαντικά καλύτερα αποτελέσματα.

Παράμετρος q_0

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου q_0 ίσες με 0,3 και 0,6. Και οι δύο τιμές που δοκιμάζονται δημιουργούν χειρότερη λύση από την αρχική.

Try	Best	Iteration	Time	Tot.Time
2	**Best: 36549**	**Iterations: 1**	**Time 0.75**	**Tot.time 10.03**
8	Best: 36581	Iterations: 1	Time 0.75	Tot.time 10.43
9	Best: 36651	Iterations: 1	Time 0.79	Tot.time 10.39

Πίνακας 21. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με q_0 ίσο με 0,3.

Try	Best	Iteration	Time	Tot.Time
8	**Best: 32687**	**Iterations: 1**	**Time 0.71**	**Tot.time 10.64**
2	Best: 32910	Iterations: 1	Time 0.69	Tot.time 10.61
4	Best: 32914	Iterations: 1	Time 0.71	Tot.time 10.64

Πίνακας 22. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με q_0 ίσο με 0,6.

Παράμετρος ρ

Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμή της παραμέτρου ρ ίση με 0,5.

Try	Best	Iteration	Time	Tot.Time
8	**Best: 30263**	**Iterations: 1**	**Time 0.64**	**Tot.time 10.38**
3	Best: 30328	Iterations: 1	Time 0.63	Tot.time 10.26
4	Best: 30368	Iterations: 1	Time 0.64	Tot.time 10.19

Πίνακας 23. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με ρ ίσο με 0,5.

Η λύση που δημιουργεί η αύξηση της παραμέτρου είναι οριακά χειρότερη από την αρχική που έχει τιμή 30188.

Παράμετρος m

Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι $1/10$, $1/2$, $1/1$ *αλλά* σαν αρχική επιλέγεται η $m=n$. Παρακάτω παρουσιάζονται τα αποτελέσματα για τιμές της παραμέτρου m ίσες με 100 και 500.

Try	Best	Iteration	Time	Tot.Time
6	**Best: 29959**	**Iterations: 5**	**Time 0.31**	**Tot.time 10.03**
5	Best: 30142	Iterations: 1	Time 0.09	Tot.time 10.00
4	Best: 30305	Iterations: 10	Time 0.51	Tot.time 10.03

Πίνακας 24. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με m ίσο με 100.

Try	Best	Iteration	Time	Tot.Time
9	**Best: 29834**	**Iterations: 1**	**Time 0.28**	**Tot.time 10.18**
6	Best: 29905	Iterations: 2	Time 0.57	Tot.time 10.12
4	Best: 30035	Iterations: 1	Time 0.29	Tot.time 10.05

Πίνακας 25. Αποτελέσματα για το πρόβλημα με 1000 Πόλεις με m ίσο με 500.

Σχεδόν όλες οι λύσεις που δημιουργούνται και για τις δύο νέες τιμές της παραμέτρου είναι καλύτερες της αρχικής. Η σημαντική βελτίωση όμως παρατηρείται στο χρόνο που ο αλγόριθμος δημιουργεί τη νέα καλύτερη λύση για το πρόβλημα των 1000 πόλεων. Οι τρείς καλύτερες λύσεις χρειάζονται λιγότερο από 1 δευτερόλεπτο για δημιουργηθούν σε σχέση με τα σχεδόν 4 δευτερόλεπτα που χρειάζεται η δημιουργία της καλύτερης λύσης με τις αρχικές παραμέτρους.

8.6 Συμπεράσματα

Τα τρία προβλήματα για 100, 500 και 1000 πόλεις ακολουθούν το TSPLIB EUC_2D format και αποτελούν τα προβλήματα αναφοράς. Για κάθε ένα από αυτά δοκιμάζεται ο αλγόριθμος με μια σειρά αρχικών παραμέτρων, που αφορούν την διαδικασία κατασκευής λύσεων και την διαδικασία ανανέωσης της φερομόνης.

Ο νέος αλγόριθμος αντλεί έμπνευση από το πλαίσιο αλγορίθμων Ant Colony Optimization και προτείνει μια νέα διαδικασία ανανέωσης της φερομόνης με σκοπό τη ταχύτερη και καλύτερη απόδοση. Συνεπώς, πλήθος τιμών μπορούν να δοκιμαστούν κατά την εφαρμογή του αλγορίθμου αν και σαν αρχικές τιμές για τις παραμέτρους της διαδικασίας χρησιμοποιούνται τιμές κοντά στις πειραματικά προτεινόμενες από τη διεθνή βιβλιογραφία.

Οι αρχικές τιμές που λαμβάνουν οι παράμετροι του αλγορίθμου δημιουργούν από τα τρία προβλήματα αναφοράς τις αρχικές λύσεις. Τα αποτελέσματα για κάθε πρόβλημα αποτελούν ένα εφαλτήριο για τις επόμενες δοκιμές του αλγορίθμου με νέες παραμέτρους ανά στιγμιότυπο του προβλήματος. Οι παράμετροι που δοκιμάζονται επηρεάζουν διαφορετικά τμήματα της λειτουργίας του αλγορίθμου, όπως για παράδειγμα οι παράμετροι οι β και q_0 που είναι ανήκουν στη διαδικασία κατασκευής λύσεων ή η παράμετρος ρ που επηρεάζει τη διαδικασία ανανέωσης της φερομόνης.

Σαν προτεινόμενο διάστημα τιμών για την παράμετρο β συναντάμε στη διεθνή βιβλιογραφία το [2,5]. Στη παρούσα εργασία η αρχική τιμή της παραμέτρου β είναι 2, αλλά και στα τρία προβλήματα τιμές ίσες με 3,5 και 5 δίνουν καλύτερα αποτελέσματα. Η παράμετρος q_0 τίθεται αρχικά ίση με 0,9 και είναι τελικά η τιμή που αποδίδει καλύτερες λύσεις και στα τρία προβλήματα.

Η παράμετρος ρ έχει ως σκοπό την ενίσχυση των λύσεων υψηλής ποιότητας και την απαλοιφή των λύσεων χαμηλής ποιότητας μέσω της απεικόνισης τους στη μνήμη της αποικίας, τον πίνακα φερομόνης. Η αρχική τιμή της παραμέτρου είναι 0,1 και στη παρούσα εργασία λαμβάνει και την τιμή 0,5. Τα αποτελέσματα όμως διαφοροποιούνται ανάλογα με το πρόβλημα, καθώς για το πρόβλημα των 100 πόλεων απέδωσε καλύτερα η αρχική τιμή, ενώ για τα άλλα δύο προβλήματα υπήρξε είτε οριακή βελτίωση, είτε οριακή χειροτέρευση των λύσεων σε σχέση με την αρχική τιμή.

Στη πλειοψηφία των υλοποιήσεων του πλαισίου ACO ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων. Είναι δηλαδή $m=n$, με την εξαίρεση του Ant Colony System όπου προτείνεται $m=10$. Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι 1/10, 1/2, 1/1 αλλά σαν αρχική επιλέγεται η $m=n$. Για το πρόβλημα με τις 100 πόλεις η τιμή της παραμέτρου ίση με 1/10, δηλαδή ίση με 10, απέδωσε καλύτερα. Όταν όμως η παράμετρος πήρε την τιμή 50 δεν υπήρξε βελτίωση. Για το πρόβλημα με τις 500 και για το πρόβλημα με τις 1000 πόλεις η αρχική τιμή δεν είναι η καλύτερη καθώς δημιουργούνται σαφώς καλύτερες λύσεις και πιο γρήγορα με τιμές ίσες με το 1/10 και το 1/2 του αριθμού των πόλεων.

Ένας αλγόριθμος της οικογένειας Ant Colony Optimization είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν. Η ρύθμιση αυτού του μηχανισμού εξαρτάται από τα διαφορετικά τμήματα που τον απαρτίζουν καθώς και τις παραμέτρους που έχει το κάθε τμήμα. Η μακροσκοπική συμπεριφορά του αλγορίθμου για παράδειγμα, εξαρτάται είτε από τον τρόπο που τα μυρμήγκια κατασκευάζουν τις λύσεις τους με βάση τον πιθανοκρατικό κανόνα μετάβασης, είτε από την διαδικασία ανανέωσης της φερομόνης πάντα σε σχέση με το υπό έρευνα πρόβλημα. Επειδή λοιπόν αποτελούνται από σύνθετα τμήματα που αλληλεπιδρούν είναι δύσκολο να προβλέψουμε την συμπεριφορά τους σε καινοτόμα προβλήματα. Ο νέος αλγόριθμος δείχνει τη σημασία του σωστού σχεδιασμού, της σωστής ρύθμισης των παραμέτρων για την επίτευξη του βέλτιστου αποτελέσματος καθώς και την προσαρμοστικότητα του πλαισίου αλγορίθμων μυρμηγκιών.

Κεφάλαιο 9

Συμπεράσματα

9.1 Σύνοψη και Συμπεράσματα

Προβλήματα Συνδυαστικής Βελτιστοποίησης

Η παρούσα διατριβή μελετά δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης και προτείνει λύσεις με μεταευριστικούς αλγορίθμους τεχνητής ευφυΐας σμήνους. Τα προβλήματα συνδυαστικής βελτιστοποίησης ασχολούνται με την αναζήτηση μεγίστης ή ελάχιστης από ένα πλήθος επιλογών που υπάρχουν στο χώρο αναζήτησης του προβλήματος. Τα προβλήματα βελτιστοποίησης μπορούν να εκφραστούν ως προβλήματα απόφασης αν θέσουμε κάποιον όρο και συνήθως εάν υπάρχει ένας γρήγορος αλγόριθμος για ένα πρόβλημα απόφασης, είναι σχετικά εύκολο να λυθεί και το αντίστοιχο πρόβλημα βελτιστοποίησης. Όμως τέτοιου είδους προβλήματα γίνονται εξαιρετικά δύσκολα όταν συμβαίνει η συνδυαστική έκρηξη.

Η συνδυαστική έκρηξη είναι η εκθετική αύξηση του μεγέθους του χώρου αναζήτησης του προβλήματος και είναι και ο κύριος λόγος που αυτά τα προβλήματα με τη συνήθως μικρή και κομψή μαθηματική έκφραση είναι τόσο δύσκολα να λυθούν και συχνά κατατάσσονται στη κλάση NP (Non deterministic Polynomial). Αυτό σημαίνει ότι τέτοιου είδους προβλήματα μπορούν να εξακριβωθούν (verified) αλλά όχι να λυθούν σε πολυωνυμικό χρόνο. Συνεπώς η κλάση NP είναι η κλάση προβλημάτων απόφασης για την οποία είναι εύκολο να *ελέγξουμε* (check) την ορθότητα της υποτιθέμενης λύσης, με τη βοήθεια της πληροφορίας που προέρχεται από το πιστοποιητικό. Δηλαδή δε ζητάμε τρόπο *εύρεσης* της λύσης αλλά μόνο τρόπο εξακρίβωσης (verify) ότι η υποτιθέμενη λύση είναι πράγματι αληθείς. Οι παραδοσιακές μαθηματικές μέθοδοι προσφέρουν τη εξασφάλιση της απόλυτα βέλτιστης λύσης αλλά όταν το μέγεθος του προβλήματος μεγαλώνει, η συνδυαστική έκρηξη στο χώρο αναζήτησης λύσεων οδηγεί αυτού του είδους τους αλγορίθμους να έχουν απαιτήσεις υπολογιστικού χρόνου που δεν είναι πρακτικά εφαρμόσιμες. Επιπρόσθετα, οι παραδοσιακές μέθοδοι χρειάζονται πολλές φορές περιορισμούς που απλοποιούν τόσο το πρόβλημα ώστε πλέον δεν ανταποκρίνονται στη ουσία του πραγματικού προβλήματος.

Για παράδειγμα, στο πρόβλημα του περιοδεύοντος πωλητή ένα πιστοποιητικό είναι μια διαδρομή (tour) της οποίας το συνολικό μήκος είναι $\leq K$. Ο αλγόριθμος A θα εξακριβώσει (verify) ότι πράγματι η διαδρομή επισκέπτεται όλες τις πόλεις και πράγματι έχει συνολικό μήκος $\leq K$.

Το πρόβλημα του περιοδεύοντος πωλητή έχει απασχολήσει πλήθος ερευνητών λόγο της απλής έκφρασης του αλλά και της μεγάλης εφαρμογής του, όπως και το σχετικά συναφές από εννοιολογική άποψη πρόβλημα δρομολόγησης οχημάτων . Επίσης το πρόβλημα τετραγωνικής ανάθεσης και η σχέση του με τις διαδικασίες παραγωγής έχουν προσελκύσει πολλές ερευνητικές προσπάθειες. Πραγματικά προβλήματα με πλήθος εφαρμογών μπορούν να αναχθούν σε αυτά τα προβλήματα της κλάσης NP Complete και οι τροποποιήσεις τους με σκοπό να καλύψουν μεγαλύτερο εύρος εφαρμογών δείχνουν το ενδιαφέρον που έχουν προκαλέσει στη ερευνητική κοινότητα καθώς και την αντίστοιχη προσπάθεια που έχει γίνει για την επίλυση τους. Μέχρι να υπάρξει σαφής απόδειξη για το αν η κλάση P είναι ίση ή διάφορη της NP , τα προβλήματα αυτά θα παραμένουν ουσιαστικά δίχως ακριβή λύση και θα συνεχίζουμε να τα αντιμετωπίζουμε με προσεγγιστικούς αλγορίθμους.

Ευριστικές και Μεταευριστικές Μέθοδοι

Οι ευριστικές και μεταευριστικές μέθοδοι έχουν σαν στόχο τη προσέγγιση βέλτιστων λύσεων σε δύσκολα προβλήματα συνδυαστικής βελτιστοποίησης με σημαντικότατα οφέλη στον υπολογιστικό χρόνο. Το οικοσύστημα των μεταευριστικών μεθόδων είναι πλούσιο και πολυσχιδές. Το κοινό σημείο όλων των μεθόδων είναι η προσπάθεια να αποφευχθεί η δημιουργία λύσεων χαμηλής ποιότητας με τη χρήση μηχανισμών που επεκτείνουν τις δυνατότητες αλγορίθμων που έχουν σχεδιαστεί για τη λύση συγκεκριμένων προβλημάτων είτε αυτοί είναι κατασκευαστικοί αλγόριθμοι είτε αλγόριθμοι τοπικής έρευνας. Τα στοιχεία διαφοροποίησης των μεταευριστικών μεθόδων έχουν να κάνουν με τις τεχνικές που χρησιμοποιούν για να αποφύγουν τον εγκλωβισμό σε τοπικά βέλτιστα και τον τρόπο που αυτές υλοποιούνται κατά τη δημιουργία των λύσεων.

Οι βασικές διαφορές μεταξύ των παραδοσιακών μαθηματικών μεθόδων και των μεταευριστικών για προβλήματα συνδυαστικής βελτιστοποίησης βρίσκονται στη ποιότητα της λύσης και στον υπολογιστικό χρόνο που χρειάζεται για να παραχθεί αυτή η λύση. Οι παραδοσιακές μαθηματικές μέθοδοι προσφέρουν τη εξασφάλιση της απόλυτα βέλτιστης λύσης αλλά όταν το μέγεθος του προβλήματος μεγαλώνει, η συνδυαστική έκρηξη στο χώρο αναζήτησης λύσεων οδηγεί αυτού του είδους τους αλγορίθμους να έχουν απαιτήσεις υπολογιστικού χρόνου που δεν είναι πρακτικά εφαρμόσιμες. Ένα ενδιαφέρων χαρακτηριστικό των μεταευριστικών μεθόδων είναι ότι δεν υπάρχει κάποιο αντικειμενικό κριτήριο για τον τερματισμό του αλγορίθμου, αλλά συνήθως χρησιμοποιείται ένας πρακτικός κανόνας ανάλογα με τη μέθοδο. Τέτοιοι κανόνες μπορεί να είναι ο αριθμός των επαναλήψεων, ο υπολογιστικός χρόνος της CPU ή ένας μέγιστος αριθμός λύσεων. Για παράδειγμα, ο αλγόριθμος Tabu Search σταματά όταν είναι άδειο το σύνολο των δυνατών λύσεων σε μια γειτονιά που ερευνά και ο αλγόριθμος Προσομοιωμένης Ανόπτησης σταμάτα με βάση το χρονοπρόγραμμα ψύξης. Επιπρόσθετα, οι παραδοσιακές μέθοδοι χρειάζονται πολλές φορές περιορισμούς που απλοποιούν τόσο το πρόβλημα ώστε πλέον δεν ανταποκρίνονται στη ουσία του πραγματικού προβλήματος. Οι μεταευριστικοί αλγόριθμοι έχουν το πλεονέκτημα της εύκολης προσαρμογής στα δεδομένα του υπό έρευνα προβλήματος αλλά δεν πρέπει να τους αντιμετωπίζουμε σαν μια λύση με εφαρμογή σε όλα τα προβλήματα. Είναι περισσότερο ένα σύνολο ιδεών χωρισμένων σε οικογένειες αλγορίθμων που το πλεονέκτημα της προσαρμοστικότητας και της γρήγορης παραγωγής λύσεων πολύ κοντά στην απόλυτα βέλτιστη τους κάνει ιδιαίτερα ενδιαφέροντες για έρευνα σε εφαρμογές δύσκολων προβλημάτων.

Η Κοινωνική Συμπεριφορά των Εντόμων και η Βιολογική Προέλευση των Ιδεών

Τα πρώτα πειράματα από το χώρο της βιολογίας για τον τρόπο αναζήτησης τροφής των μυρμηγκιών, μέσω των ιδεών της αυτο-οργάνωσης και της στιγμεργείας, έδειξαν τις πρώτες αρχές για το σχηματισμό πολύπλοκων λειτουργικών δομών σε συλλογικό επίπεδο αποικίας.

Η Ευφυΐα Σμήνους είναι τεχνητή ευφυΐα που βασίζεται στη συλλογική συμπεριφορά αποκεντρωμένων, αυτο-οργανομένων συστημάτων. Ο πληθυσμός των ατόμων του συστήματος αποτελείται από άτομα που είτε είναι όμοια είτε διαφέρουν ελάχιστα μεταξύ τους αλλά είναι σχετικά απλά σαν μονάδες. Αυτό ο πληθυσμός αλληλεπιδρά δίχως να υπάρχει ένα γενικό πλαίσιο κανόνων ή συνειδητή συμπεριφορά των ατόμων και το σύνολο των αλληλεπιδράσεων εκφράζει τη συλλογική ευφυΐα του συστήματος. Αυτό είναι απόρροια της τοπικής αλληλεπίδρασης ατόμων που γειτνιάζουν στο χώρο και της συμπεριφοράς τους που ορίζεται από απλούς κανόνες. Συχνά αυτή η συμπεριφορά του κάθε ατόμου περιγράφεται με στοχαστικούς όρους που βασίζονται στη τοπική αντίληψη του ατόμου για το περιβάλλον του. Η Ευφυΐα Σμήνους περιλαμβάνει φυσικά και τεχνητά συστήματα που αποτελούνται από πολλά άτομα που συντονίζονται με αποκεντρωμένο έλεγχο και αυτο-οργάνωση. Συγκεκριμένα εστιάζει στις συλλογικές συμπεριφορές που αναδύονται από τις τοπικές αλληλεπιδράσεις των ατόμων μεταξύ τους και με το περιβάλλον. Παραδείγματα τέτοιων συστημάτων είναι οι αποικίες μυρμηγκιών και τερμιτών, σμήνη πουλιών, κοπάδια ψαριών, αγέλες ζώων και αποικίες μελισσών. Ανθρώπινες εφαρμογές με τέτοιες συμπεριφορές είναι πολύ-ρομποτικά συστήματα και λογισμικό με σκοπό τη βελτιστοποίηση.

Τα τέσσερα βασικά στοιχεία της αυτο-οργάνωσης, όπως η θετική ανάδραση, αρνητική ανάδραση, ενίσχυση τυχαίων διακυμάνσεων, αλλά και η ύπαρξη αλληλεπιδράσεων μεταξύ των ατόμων της αποικίας, δημιουργούν ένα σύστημα όπου το μεμονωμένο έντομο μπορεί να διεκπεραιώσει μόνο απλές λειτουργίες και δε λαμβάνεται υπόψη η περίπλοκη δομή του μεμονωμένου εντόμου. Η στιγμεργία είναι ένας αποκεντρωμένος μηχανισμός ροής πληροφοριών στο οποίο τα άτομα, μέρη του συστήματος, επικοινωνούν μέσω των αλλαγών που προκαλούν στο τοπικό τους περιβάλλον και στα κοινωνικά έντομα όπως τα μυρμήγκια υλοποιείται με την εναπόθεση στο έδαφος της χημικής ουσίας που ονομάζεται φερομόνη.

Οι παραπάνω ιδέες δημιουργούν ένα αρχικό πλαίσιο προσέγγισης για την κατανόηση αυτού δίχως κεντρική διοίκηση συστήματος των κοινωνικών εντόμων, που καταφέρνει να λειτουργεί αρμονικά σαν σύνολο ανακαλύπτοντας τη συντομότερη διαδρομή προς την πηγή τροφής. Τα πειράματα με διπλή γέφυρα αναλύουν την εξερευνητική συμπεριφορά των μυρμηγκιών για την αναζήτηση τροφής και αποκαλύπτουν το ρόλο της φερομόνης στην επιλογή της διαδρομής προς την τροφή. Η μοντελοποίηση της εξερευνητικής συμπεριφοράς εκφράζεται στις πρωτόλειες εξισώσεις που δίδονται στις αρχές της δεκαετίας του 90 και αποτελούν την αφετηρία για τις περεταίρω μελέτες που τελικά έφεραν το ψηφιακό ανάλογο του μοντέλου αναζήτησης τροφής στις αποικίες μυρμηγκιών.

Η συνάφεια μεταξύ βιολογικού και υπολογιστικού μοντέλου ενισχύεται από το γεγονός ότι και στα δύο υπάρχει μια αποικία συνεργαζόμενων ατόμων, καθώς και ψηφιακή απεικόνιση του ίχνους φερομόνης, και μέσω αυτού της στιγμεργίας. Επίσης η αναζήτηση ελάχιστης διαδρομής, οι τοπικές κινήσεις *αλλά* και οι στοχαστικές και μυωπικές πολιτικές μετάβασης αποτελούν στοιχεία ομοιότητας μεταξύ των δύο μοντέλων.

Σαν αντίλογος για το ψηφιακό υπολογιστικό μοντέλο αντιτάσσεται το διακριτό περιβάλλον του ψηφιακού μοντέλου εναντίον του συνεχούς περιβάλλοντος του βιολογικού αλλά και η απλότητα του προβλήματος που συναντάται στη φύση. Η σχετικότητα του περιβάλλοντος ως επιχείρημα, έχει σκοπό να αναδείξει ότι πιθανώς οι αναλογίες που χρησιμοποιούνται για την απεικόνιση των βιολογικών φαινομένων δεν είναι οι καλύτερες δυνατές. Το κύριο επιχείρημα για τη διαφορά μεταξύ των δύο μοντέλων έχει να κάνει με την εφαρμογή της στρατηγικής επικοινωνίας και την κατανεμημένη φύση του προβλήματος καθώς οι τομείς αυτοί αντιμετωπίζονται με σχετικά απλή προσέγγιση.

Με την ωρίμανση της μαθηματικής απεικόνισης αυτών των ιδεών γεννήθηκαν οι αρχικοί αλγόριθμοι μυρμηγκιών και παρουσιάστηκαν οι πρώτοι απτοί αλγόριθμοι που αντλούσαν την έμπνευση για τον τρόπο λειτουργίας τους από τον τρόπο λειτουργίας των κοινωνικών εντόμων που παρατηρήθηκε στις αποικίες μυρμηγκιών.

Η οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών

Τα κύρια σημεία ενός αλγορίθμου μυρμηγκιών είναι : ο πίνακας φερομόνης, ο πιθανοκρατικός κανόνας κατασκευής λύσεων, ο κανόνας ανανέωσης της φερομόνης, η ευριστική πληροφορία και η προαιρετική ενσωμάτωση daemon action συνήθως με τη μορφή μιας μεθόδου τοπικής έρευνας.

Ο πίνακας φερομόνης είναι η μνήμη του αλγορίθμου και είναι η απεικόνιση της συλλογικής εμπειρίας της αποικίας για το περιβάλλον της. Αποτελείται από τα στοιχεία που χρειάζονται για την κατασκευή των λύσεων. Για παράδειγμα στο πρόβλημα του περιοδεύοντος πωλητή είναι ένας πίνακας του οποίου οι γραμμές και στήλες αντιστοιχούν στις συνδέσεις μεταξύ των πόλεων και οι τιμές του σχετίζονται με την πιθανότητα επιλογής κάθε πόλης σαν επόμενο κόμβο στη λύση.

Ο πιθανοκρατικός κανόνας κατασκευής λύσεων συνδυάζει την ευριστική πληροφορία και τις πληροφορίες που απεικονίζονται στο πίνακα φερομόνης για να επιλεγεί ο επόμενος κόμβος που θα προστεθεί στη τρέχουσα λύση. Ο κανόνας ανανέωσης της φερομόνης είναι ο τρόπος με τον οποίο οι πληροφορίες από την τρέχουσα επανάληψη του αλγορίθμου ενσωματώνονται στο πίνακα φερομόνης. Με αυτόν τον τρόπο η αποικία απεικονίζει τη γνώση που αποκτά για το περιβάλλον της. Η ευριστική πληροφορία και η προαιρετική ενσωμάτωση daemon action συνήθως με τη μορφή μιας μεθόδου τοπικής έρευνας είναι στην ευχέρεια του σχεδιαστή του αλγορίθμου και οι επιλογές που γίνονται έχουν σημαίνον ρόλο στη λειτουργία του αλγορίθμου.

Για να κατασκευαστεί μια λύση τα τεχνητά μυρμήγκια κινούνται στην αναπαράσταση του προβλήματος υπό τη μορφή ενός γραφήματος, όπου κάθε κορυφή είναι ένα πιθανό μέρος της λύσης. Ο πιθανοκρατικός κανόνας κατασκευής λύσεων χρησιμοποιείται για την επιλογή της επόμενης κορυφής που θα επισκεφθεί το μυρμήγκι. Ο αλγόριθμος σταματά όταν εξαντληθούν οι κορυφές του γραφήματος ή έχει πληρωθεί κάποιο κριτήριο τερματισμού. Πρέπει να σημειωθεί ότι δημιουργείται πλήθος λύσεων καθώς κάθε μυρμήγκι κατασκευάζει μια λύση. Αυτές ανάλογα με το αν έχει ενσωματωθεί κάποια προαιρετική daemon action εκτιμούνται και ύστερα, ο κανόνας ανανέωσης της φερομόνης προσθέτει τις πληροφορίες για τα τμήματα της καλύτερης λύσης στον πίνακα φερομόνης. Συνήθως επιλέγεται η καλύτερη λύση της τρέχουσας επανάληψης για να χρησιμοποιηθεί από τον κανόνα ανανέωση της φερομόνης .

Η μέθοδος Ant Colony Optimization εκφράζει το σχετικά πρόσφατο ερευνητικό πεδίο των αλγορίθμων τεχνητής ευφυΐας σμήνους. Είναι μια οικογένεια αλγορίθμων που υλοποιεί τις ιδέες των συστημάτων τεχνητής ευφυΐας σμήνους, όπου οι αρχές της θετικής/αρνητικής ανάδρασής, της ενίσχυσης τυχαίων διακυμάνσεων, της ύπαρξης αλληλεπιδράσεων μεταξύ των ατόμων της αποικίας καθώς και η στιγμεργετική επικοινωνία, με εκφράζονται σε ένα πλαίσιο με σκοπό τη βέλτιστη και ταχύτερη λύση.

Το προφανές συμπέρασμα σχετικά με την κατανόηση των λειτουργίας των αλγορίθμων ACO είναι ότι ένας αλγόριθμος της οικογένειας Ant Colony Optimization είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν. Η ρύθμιση αυτού του μηχανισμού εξαρτάται από τα διαφορετικά τμήματα που τον απαρτίζουν καθώς και τις παραμέτρους που έχει το κάθε τμήμα. Η μακροσκοπική συμπεριφορά του αλγορίθμου για παράδειγμα, εξαρτάται είτε από τον τρόπο που τα μυρμήγκια κατασκευάζουν τις λύσεις τους με βάση τον πιθανοκρατικό κανόνα μετάβασης, είτε από τη διαδικασία ανανέωσης της φερομόνης πάντα σε σχέση με το υπό έρευνα πρόβλημα. Επειδή λοιπόν αποτελούνται από σύνθετα τμήματα που αλληλεπιδρούν είναι δύσκολο να προβλέψουμε τη συμπεριφορά τους στο κάθε προβλήματα. Γενικά για μια εφαρμογή στους αλγορίθμους ACO, η δυσκολία έγκειται στο προσδιορισμό του σωστού σχεδιασμού και στη ρύθμιση του πλήθους των παραμέτρων με σκοπό τη βέλτιστη απόδοση στο υπό έρευνα πρόβλημα. Η επιλογή της σωστής περιοχής τιμών των παραμέτρων είναι ζήτημα πειραματισμού με την εφαρμογή.

Συνοπτικά, παρατηρούμε ότι οικογένεια Μεταευριστικών Αλγορίθμων Βελτιστοποίησης Μυρμηγκιών ACO διαθέτει αρκετά χαρακτηριστικά που όταν συνδυάζονται δημιουργούν καινοτόμες προσεγγίσεις : χρησιμοποιεί ένα *πληθυσμό* (αποικία) τεχνητών μυρμηγκιών που *κατασκευάζουν* λύσεις εκμεταλλευόμενα την *έμμεση μορφή μνήμης* που είναι οι *τεχνητές φερομόνες*.

9.2 Εφαρμογές Μεταευριστικών Μεθόδων σε Προβλήματα Συνδυαστικής Βελτιστοποίησης

Η οικογένεια μεταευριστικών αλγορίθμων μυρμηγκιών (Ant Colony Optimization) ανήκει στο σχετικά πρόσφατο ερευνητικό πεδίο των στοχαστικών μεταευριστικών μεθόδων όπως η προσομοιωμένη ανόπτηση, η tabu search, και οι γενετικοί αλγόριθμοί, που βασίζονται σε αρχές που προέρχονται από την παρατήρηση φυσικών φαινομένων. Η τελική οικογένεια αλγορίθμων με σκοπό την πρακτική τους χρήση, απέχουν αρκετά από το φυσικό μοντέλο που την ενέπνευσε. Συχνά οι αλγόριθμοι ACO έχουν εμπλουτιστεί με λειτουργίες που δεν έχουν αντίστοιχο στα πραγματικά μυρμήγκια, όπως για παράδειγμα δράσεις που επιλέγονται συνειδητά από την συλλογική συμπεριφορά των μυρμηγκιών όπως το 'άριστο' μυρμήγκι στο αλγόριθμο Max-Min Ant System. Είναι μια μέθοδος ιδιαίτερα επιτυχημένη καθώς έχει εφαρμοστεί σε πλήθος NP-

Hard προβλημάτων καθώς οι αρχές της στιγμεργετικής επικοινωνίας αποδεικνύουν πειραματικά την αποτελεσματικότητα τους.

Πρόβλημα Περιοδεύοντος Πωλητή

Η μέθοδος της Προσομοιωμένης Ανόπτησης είναι ιδιαίτερα επιτυχημένη και υπάρχουν πολλές παραλλαγές του αλγορίθμου. Η σχέση των Αλγορίθμων Μυρμηγκιών και της Προσομοιωμένης Ανόπτησης είναι διττή. Η μέθοδος ανταγωνίζεται τους Αλγορίθμους Μυρμηγκιών όταν εφαρμόζεται μόνη της, αλλά μπορεί να χρησιμοποιηθεί και μαζί με την Π.Α. να λειτουργεί σαν μέθοδος τοπικής έρευνας (local search). Η βασική διαφορά τους είναι ότι οι αλγόριθμοι μυρμηγκιών έχουν τον πίνακα φερομόνης που λειτουργεί ως μνήμη του αλγορίθμου ενώ η προσομοιωμένη ανόπτηση δεν κρατά κάποιο αρχείο παλιών λύσεων.

Η προσπάθεια να ελαχιστοποιηθεί το συνολικό κόστος μετάβαση για δεκατρία λιμάνια, λαμβάνοντας υπόψη το κόστος των καυσίμων και χρήσης των λιμανιών, με τη χρήση των δύο αλγορίθμων δείχνει την ικανότητα της πρότασης να ανακαλύπτει γρήγορα ποιοτικές λύσεις. Επίσης το εγγενές χαρακτηριστικό της αποφυγής εγκλωβισμού σε τοπικά βέλτιστα, καθώς είναι δυνατό να μεταβεί και σε χειρότερη λύση σε κάποια επανάληψη, τον διαφοροποιεί σε σχέση με άλλους αλγορίθμους. Ο κύριος ρυθμιστικός παράγοντας της συμπεριφοράς του αλγορίθμου είναι το χρονοπρόγραμμα ψύξης. Ένα σχετικά γρήγορο χρονοπρόγραμμα ψύξης, όπως και στο φυσικό μοντέλο ανόπτησης μετάλλων, δεν θα δώσει στα μόρια/τμήματα λύσης το χρόνο που χρειάζονται ώστε να διαταχθούν με την ελάχιστη ενέργεια/κόστος. Όμως, ένα πολύ αργό πρόγραμμα ψύξης θα αυξήσει δυσανάλογα τον υπολογιστικό χρόνο που χρειάζεται για την παραγωγή λύσεων. Σημαντικό επίσης είναι το κριτήριο αποδοχής νέων λύσεων καθώς σε αυτό το σημείο κρίνεται η επιλογή μεταξύ εξερεύνησης νέων λύσεων και εκμετάλλευσης παλαιότερων. Αποτελέσματα υψηλής ποιότητας βασίζονται στη σωστή επιλογή αυτών των κριτηρίων στο υπό έρευνα πρόβλημα και ουσιαστικά από αυτά εξαρτάται η ευρωστία του αλγορίθμου.

Προβλήματα Δρομολόγησης

Η επιλογή των παραμέτρων του αλγορίθμου, όπως το αρχικό επίπεδο φερομόνης, είναι ιδιαίτερα σημαντική για τη σύγκλιση του αλγορίθμου καθώς παρατηρούνται σημαντικές διακυμάνσεις των αποτελεσμάτων τροποποιώντας ελάχιστα τις αρχικές ποσότητες φερομόνης. Στον αλγόριθμο Ant Colony System διαδικασία ανανέωσης της φερομόνης επηρεάζει και την σχετική απόδοση του αλγορίθμου. Στη συγκεκριμένη υλοποίηση, η διαδικασίας ανανέωσης φερομόνης, δίνει έμφαση στο ορισμό της αρχική ποσότητας φερομόνης τ_0. Είναι προφανές ότι η ποιότητα των αποτελεσμάτων είναι άμεσα συναρτώμενη των αρχικών τιμών της φερομόνης καθώς παρατηρούνται σημαντικές διακυμάνσεις των αποτελεσμάτων τροποποιώντας τις αρχικές ποσότητες φερομόνης. Δηλαδή με βάση τα δεδομένα του προβλήματος, η υψηλή τιμή 4000 δεν ενίσχυσε την εξερεύνηση (exploration) νέων λύσεων, παρά την ύπαρξη του συντελεστή εξάτμισης 'ρ', αλλά πρότεινε την εκμετάλλευση (exploitation) παλαιών μη βέλτιστων λύσεων.

Αξιοσημείωτο είναι και το γεγονός ότι η αρχική τιμή φερομόνης ίση με 50 δεν επέτρεψε στους μηχανισμούς στιγμεργείας, μέσω θετικής ανάδρασης, να σχηματίσουν την βέλτιστη διαδρομή. Προφανώς για το συγκεκριμένο πρόβλημα και δεδομένα, ο ρυθμός εξάτμισης επικρατούσε πολύ γρήγορα και δεν επέτρεπε το σχηματισμό ίχνους φερομόνης που θα στρατολογούσε και άλλα μυρμήγκια. Τελικά, με αρχική τιμή φερομόνης 500 ελαχιστοποιούμε την αντικειμενική συνάρτηση του προβλήματος. Σημαντικό είναι το γεγονός ότι οι παράμετροι 'β' και 'ρ' παρέμειναν σταθεροί κατά την υλοποίηση με σκοπό η όλη συμπεριφορά του αλγορίθμου να εξαρτάται από την αρχική ποσότητα της φερομόνης η οποία ορίζει την ευελιξία του συστήματος στην εύρεση νέων, κάθε φορά, λύσεων.

Επίσης, παρατηρείται ότι μεγαλύτερος αριθμός επαναλήψεων του αλγορίθμου 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 γύρω από ένα σημείο. Στο σημείο αυτό, είναι πολύ δύσκολο (αν και όχι απίθανο) να βρεθεί μία σημαντικά καλύτερη λύση και λαμβάνεται η χαμηλότερη τιμή που έχει βρεθεί μέχρι τη στιγμή του τερματισμού ως η βέλτιστη.

Η εφαρμογή του μεταευριστικού πλαισίου ACO βασίζεται στο σωστό σχεδιασμό του αλγορίθμου και στη ρύθμιση των παραμέτρων σε σχέση με το υπό έρευνα πρόβλημα. Στη συγκεκριμένη εφαρμογή του Ant Colony System σκοπός είναι η βελτιστοποίηση των θαλάσσιων μεταφορών σε ένα σύνολο 13 κόμβων πελατών με επιπρόσθετους περιορισμούς για τις απαιτήσεις των κόμβων.

Η ρύθμισή των παραμέτρων είναι κρίσιμη για την απόδοση του αλγορίθμου και η ανακάλυψη μιας περιοχής τιμών που δίνουν τα βέλτιστα βασίζεται σε πειραματικές επαναλήψεις της εφαρμογής. Η στάθμιση μεταξύ εξερεύνησης νέων λύσεων και εκμετάλλευσης παλαιοτέρων γίνεται από τις παραμέτρους β και q_0, που ρυθμίζουν το την κατασκευαστική συμπεριφορά του αλγορίθμου. Στην παρούσα υλοποίηση, η παράμετρος β πήρε την τιμή 1 και 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. Επίσης σημαντικό ρόλο έχει και ο παράγοντας 'ρ' της τεχνητής εξάτμισης της φερομόνης οποίος ήταν ίσος με 0.5. Σκοπός αυτού του παράγοντα είναι να εμποδίσει την υπερβολική συγκέντρωση φερομόνης στις διαδρομές, πράγμα που θα οδηγούσε σε πρόωρη σύγκλιση του αλγορίθμου σε μη βέλτιστες λύσεις.

Πρόβλημα Πρόβλημα Τοποθέτησης Κατασκευαστικών Κυψελών

Η υλοποίηση του αλγόριθμου Ant Colony System γίνεται έτσι ώστε να εφαρμοστεί στο πρόβλημα της τοποθέτησης μηχανών σε σειρά με σκοπό την εύρεση της βέλτιστης τοποθέτησης. Σημαντικά πλεονεκτήματα του μοντέλου της ελάχιστης προς τα πίσω ροής είναι η απλότητα του που κάνει εύκολο να υλοποιηθεί και το γεγονός ότι δεν έχει κόστος διότι χρησιμοποιεί δεδομένα που υπάρχουν στην παραγωγή. Όμως για την επιτυχή υλοποίηση τα δεδομένα που απαιτούνται είναι να υπάρχει σαφώς καθορισμένη διεργασία παραγωγής και σαφώς καθορισμένη ζητούμενη ποσότητα παραγωγής. Αλλαγές σε αυτούς τους παράγοντες οδηγεί σε αναδιοργάνωση της διάταξης των μηχανών. Ο τρόπος υλοποίησης και η αναφορά σε παραδοχές που σχετίζονται με το πρόβλημα της κυψελοειδούς παραγωγής δημιουργεί ένα πλαίσιο που με σχετικά λίγες αλλαγές μπορεί να εφαρμοστεί και σε άλλα προβλήματα κυψελοειδούς παραγωγής.

Πολύ σημαντικός παράγοντας στις υλοποιήσεις αλγορίθμων μυρμηγκιών είναι ο κατάλληλος προσδιορισμός των αρχικών παραμέτρων όπως η ποσότητα της φερομόνης η οποία μέσο της στιγμεργετικής επικοινωνίας θα οδηγήσει στην επιλογή των καλύτερων λύσεων έναντι των άλλων. Σε αυτή την αναζήτηση επίσης σημαντικό ρόλο παίζει και η ρύθμιση του παράγοντα 'β' στον κανόνα μετάβασης καθώς αυτός σταθμίζει τη φερομόνη και την ευριστική πληροφορία στη διαδικασία της επιλογής του επόμενου κόμβου-μηχανής. Η προσαρμοστικότητα του αλγορίθμου βασίζεται σε αυτούς τους παράγοντες.

9.3 Η Νέα Προσέγγιση του Αλγορίθμου Ant Colony System

Τα τρία προβλήματα για 100, 500 και 1000 πόλεις ακολουθούν το TSPLIB EUC_2D format και αποτελούν τα προβλήματα αναφοράς. Για κάθε ένα από αυτά δοκιμάζεται ο αλγόριθμος με μια σειρά αρχικών παραμέτρων, που αφορούν την διαδικασία κατασκευής λύσεων και την διαδικασία ανανέωσης της φερομόνης.

Ο νέος αλγόριθμος αντλεί έμπνευση από το πλαίσιο αλγορίθμων Ant Colony Optimization και προτείνει μια νέα διαδικασία ανανέωσης της φερομόνης με σκοπό τη ταχύτερη και καλύτερη απόδοση. Συνεπώς, πλήθος τιμών μπορούν να δοκιμαστούν κατά την εφαρμογή του αλγορίθμου αν και σαν αρχικές τιμές για τις παραμέτρους της διαδικασίας χρησιμοποιούνται τιμές κοντά στις πειραματικά προτεινόμενες από τη διεθνή βιβλιογραφία.

Οι αρχικές τιμές που λαμβάνουν οι παράμετροι του αλγορίθμου δημιουργούν από τα τρία προβλήματα αναφοράς τις αρχικές λύσεις. Τα αποτελέσματα για κάθε πρόβλημα αποτελούν ένα εφαλτήριο για τις επόμενες δοκιμές του αλγορίθμου με νέες παραμέτρους ανά στιγμιότυπο του προβλήματος. Οι παράμετροι που δοκιμάζονται επηρεάζουν διαφορετικά τμήματα της λειτουργίας του αλγορίθμου, όπως για παράδειγμα οι παράμετροι οι β και q_0 που είναι ανήκουν στη διαδικασία κατασκευής λύσεων ή η παράμετρος ρ που επηρεάζει τη διαδικασία ανανέωσης της φερομόνης.

Σαν προτεινόμενο διάστημα τιμών για την παράμετρο β συναντάμε στη διεθνή βιβλιογραφία το [2,5]. Στη παρούσα εργασία η αρχική τιμή της παραμέτρου β είναι 2, αλλά και στα τρία προβλήματα τιμές ίσες με 3,5 και 5 δίνουν καλύτερα αποτελέσματα. Η παράμετρος q_0 τίθεται αρχικά ίση με 0,9 και είναι τελικά η τιμή που αποδίδει καλύτερες λύσεις και στα τρία προβλήματα.

Η παράμετρος ρ έχει ως σκοπό την ενίσχυση των λύσεων υψηλής ποιότητας και την απαλοιφή των λύσεων χαμηλής ποιότητας μέσω της απεικόνισης τους στη μνήμη της αποικίας, τον πίνακα φερομόνης. Η αρχική τιμή της παραμέτρου είναι 0,1 και στη παρούσα εργασία λαμβάνει και την τιμή 0,5. Τα αποτελέσματα όμως διαφοροποιούνται ανάλογα με το πρόβλημα, καθώς για το πρόβλημα των 100 πόλεων απέδωσε καλύτερα η αρχική τιμή, ενώ για τα άλλα δύο προβλήματα υπήρξε είτε οριακή βελτίωση, είτε οριακή χειροτέρευση των λύσεων σε σχέση με την αρχική τιμή.

Στη πλειοψηφία των υλοποιήσεων του πλαισίου ACO ο αριθμός των μυρμηγκιών είναι ίσος με τον αριθμό των πόλεων. Είναι δηλαδή $m = n$, με την εξαίρεση του Ant Colony System όπου προτείνεται $m = 10$. Στη παρούσα υλοποίηση η παράμετρος m λαμβάνει τιμές βάση του κλάσματος m/n και είναι 1/10, 1/2, 1/1 αλλά σαν αρχική επιλέγεται η $m = n$. Για το πρόβλημα με τις 100 πόλεις η τιμή της παραμέτρου ίση με 1/10, δηλαδή ίση με 10, απέδωσε καλύτερα. Όταν όμως η παράμετρος πήρε την τιμή 50 δεν υπήρξε βελτίωση. Για το πρόβλημα με τις 500 και για το πρόβλημα με τις 1000 πόλεις η αρχική τιμή δεν είναι η καλύτερη καθώς δημιουργούνται σαφώς καλύτερες λύσεις και πιο γρήγορα με τιμές ίσες με το 1/10 και το 1/2 του αριθμού των πόλεων.

Κατανόηση των Λειτουργίας των Αλγορίθμων ACO

Ένας αλγόριθμος της οικογένειας Ant Colony Optimization είναι ένα σύνθετος μηχανισμός με διαφορετικά τμήματα που αλληλεπιδρούν. Η ρύθμιση αυτού του μηχανισμού εξαρτάται από τα διαφορετικά τμήματα που τον απαρτίζουν καθώς και τις παραμέτρους που έχει το κάθε τμήμα. Η μακροσκοπική συμπεριφορά του αλγορίθμου για παράδειγμα, εξαρτάται είτε από τον τρόπο που τα μυρμήγκια κατασκευάζουν τις λύσεις τους με βάση τον πιθανοκρατικό κανόνα μετάβασης, είτε από την διαδικασία ανανέωσης της φερομόνης πάντα σε σχέση με το υπό έρευνα πρόβλημα. Επειδή λοιπόν αποτελούνται από σύνθετα τμήματα που αλληλεπιδρούν είναι δύσκολο να προβλέψουμε την συμπεριφορά τους σε καινοτόμα προβλήματα.

Μια κύρια μέθοδος διαχωρισμού των μεταευριστικών μεθόδων είναι εάν ο προσανατολισμός της δράσης τους προέρχεται από κατασκευαστικές μεθόδους ή μεθόδους τοπικής έρευνας. Για παράδειγμα η οικογένεια αλγορίθμων ACO είναι σαφώς προσανατολισμένη σε κατασκευαστικές μεθόδους. Επιπρόσθετα ένα κριτήριο διαχωρισμού είναι εάν σε κάθε επανάληψη ο αλγόριθμος διαχειρίζεται μια λύση ή ένα πληθυσμό λύσεων. Οι αλγόριθμοι ACO και οι Γενετικοί Αλγόριθμοι είναι οι μόνες μεταευριστικές μέθοδοι που διατηρούν ένα πληθυσμό λύσεων και τον διαχειρίζονται ανάλογα με το σχεδιασμό του αλγορίθμου. Επίσης σημείο διαφοροποίησης είναι η χρήση μνήμης στο σχεδιασμό του αλγορίθμου. Για παράδειγμα και οι αλγόριθμοι ACO και ο αλγόριθμος Tabu Search χρησιμοποιούν μια αναπαράσταση μνήμης παλαιότερων λύσεων για να κατευθύνουν την μελλοντική αναζήτηση λύσεων. Ο αλγόριθμος Tabu Search είτε αποθηκεύει ολόκληρες λύσεις ή τμήματα λύσεων και η οι αλγόριθμοι ACO χρησιμοποιούν τον πίνακα φερομόνης για την απεικόνιση της συλλογικής εμπειρίας της αποικίας.

Οι παρατηρήσεις από τις παραπάνω εφαρμογές δίνουν συμπεράσματα που ισχύουν γενικότερα για τους μεταευριστικούς αλγορίθμους. Η νέα μορφή του αλγορίθμου ACS έδειξε ότι μπορεί να αποδώσει καθώς η ρύθμιση των παραμέτρων βοηθά στην εύρεση ποιοτικών λύσεων καθώς οδηγεί τον αλγόριθμο σε ικανοποιητική εξερευνητική συμπεριφορά. Καίριος είναι και ο ρόλος της ισορροπίας της φερομόνης και της ευριστικής πληροφορίας κατά τη διαδικασία κατασκευής λύσεων. Αφού επιλεγεί ο σωστός σχεδιασμός του αλγορίθμου από το πλαίσιο της οικογένειας ACO, σε σχέση με το υπό έρευνα πρόβλημα, η στάθμιση των προανφερθέντων παραγόντων γίνεται από τις παραμέτρους α, β, q_0 για την κατασκευή λύσεων, ρ για τη διαδικασία ανανέωσης της φερομόνης και m για τη γενικότερη απόδοση του αλγορίθμου. Με βάση και τους επιπρόσθετους περιορισμούς του κάθε προβλήματος, πρέπει λοιπόν να επιλεγούν οι τιμές που δίνουν τη βέλτιστη ισορροπία.

Γενικά για μια εφαρμογή στους αλγορίθμους ACO, η δυσκολία έγκειται στο προσδιορισμό του σωστού σχεδιασμού και στη ρύθμιση του πλήθους των παραμέτρων με σκοπό τη βέλτιστη απόδοση στο υπό έρευνα πρόβλημα. Η επιλογή της σωστής περιοχής τιμών των παραμέτρων είναι ζήτημα πειραματισμού με την εφαρμογή.

Ευριστική Πληροφορία

Η ευριστική πληροφορία είναι πολύ σημαντική όταν γίνεται εφαρμογή αλγορίθμων ACO σε προβλήματα συνδυαστικής βελτιστοποίησης όπως του Περιοδεύοντος Πωλητή (TSP) για την κατασκευή ποιοτικών λύσεων. Στην εφαρμογή των αλγορίθμων, αφού τεθεί το αρχικό επίπεδο φερομόνης, το οποίο επιλέγεται από τον ερευνητή, για τις πρώτες επαναλήψεις του αλγορίθμου τα μυρμήγκια δεν στηρίζονται στην καθοδήγηση των φερομονών και είναι πολύ πιθανό να κατασκευάσουν και να ενισχύουν λύσεις χαμηλής ποιότητας. Ο ρόλος της ευριστικής πληροφορίας είναι να αποτρέψει αυτό το γεγονός δίδοντας μια γενική στρατηγική κατεύθυνση στην έρευνα σε αυτά τα πρώτα στάδια.

Η χρήση ή μη ευριστικής πληροφορίας μπορεί εύκολα να ρυθμιστεί στους αλγορίθμους ACO θέτοντας την παράμετρο β ίση με μηδέν στον πιθανοκρατικό κανόνα μετάβασης. Η πρακτική διάσταση του θέματος όπως προκύπτει από τα πειραματικά αποτελέσματα είναι πιο περίπλοκη. Για παράδειγμα για ένα πρόβλημα σαν του Περιοδεύοντος Πωλητή (TSP) η απόσταση μεταξύ των πόλεων είναι μια προφανής και δίχως υπολογιστικό κόστος επιλογή ευριστικής πληροφορίας. Όμως σε άλλα προβλήματα ο ορισμός της ευριστικής πληροφορίας έτσι ώστε να αποδίδει με καλύτερες διαδρομές, μπορεί να κοστίζει πολύ σε υπολογιστικά βήματα. Η ρύθμιση λοιπόν της τιμής της παραμέτρου β είναι ένα σημείο ελέγχου της απόδοση του αλγορίθμου.

Εναλλακτικές Μεθοδολογίες

Οι εναλλακτικές μεθοδολογίες που παρουσιάζονται στη παρούσα εργασία είναι αλγόριθμοι οι οποίοι μπορούν να εφαρμοστούν είτε μόνες τους είτε ως μέθοδοι τοπικής έρευνας και στη διεθνή βιβλιογραφία υπάρχουν δημοσιεύσεις όπου οι μέθοδοι υλοποιούνται είτε με τον ένα είτε με τον άλλο τρόπο, αλλά όχι πάντα με επιτυχημένα αποτελέσματα. Στο συνδυασμό για παράδειγμα της Tabu Search με κάποιο αλγόριθμο μυρμηγκιών η επιλογή του σωστού σχεδιασμού και των σωστών παραμέτρων είναι κρίσιμη, πράγμα που δυσκολεύει ακόμα περισσότερο το συνδυασμό της με αλγορίθμους μυρμηγκιών.

Ένα άλλο παράδειγμα είναι η μέθοδος Προσομοιωμένης Ανόπτησης που είναι ιδιαίτερα επιτυχημένη και υπάρχουν πολλές παραλλαγές του αλγορίθμου. Η σχέση των Αλγορίθμων Μυρμηγκιών και της Προσομοιωμένης Ανόπτησης είναι διττή. Η μέθοδος ανταγωνίζεται τους Αλγορίθμους Μυρμηγκιών όταν εφαρμόζεται μόνη της, αλλά μπορεί να χρησιμοποιηθεί και μαζί με την Π.Α. να λειτουργεί σαν μέθοδος τοπικής έρευνας (local search). Η βασική διαφορά τους είναι ότι οι αλγόριθμοι μυρμηγκιών έχουν τον πίνακα φερομόνης που λειτουργεί ως μνήμη του αλγορίθμου ενώ η προσομοιωμένη ανόπτηση δεν κρατά κάποιο αρχείο παλιών λύσεων.

Η φύση του υπό έρευνα προβλήματος και ο σχεδιασμός του κάθε αλγορίθμου σχεδόν προκαθορίζουν την απόδοση μιας αλγοριθμικής εφαρμογής. Οι Γενετικοί Αλγόριθμοι και οι Αλγόριθμοι Μυρμηγκιών έχουν ένα σύνολο επιλογών στο σχεδιασμό των αλγορίθμων που είναι κοινό, όπως για παράδειγμα η αναπαράσταση του προβλήματος, το μέγεθος του πληθυσμού των μυρμηγκιών ανά επανάληψη και η χρήση ή μη μεθόδου τοπικής έρευνας. Το μεγαλύτερο βάρος όμως στο σχεδιασμό του αλγορίθμου τον θέτει το υπό έρευνα πρόβλημα και σε αυτό, οι αλγόριθμοι μυρμηγκιών έχουν το πλεονέκτημα να ταιριάζουν καλύτερα σε προβλήματα που η περιγραφή τους στηρίζεται σε γράφημα. Επίσης η κατασκευαστική φύση των αλγορίθμων μυρμηγκιών δεν χρειάζεται τελεστές μετάβασης για να δημιουργήσει νέες λύσεις.

9.4 Μελλοντική Έρευνα

Οι αλγόριθμοι μυρμηγκιών όπως περιγράφονται στο κεφάλαιο 4 βασίζονται στην μετάβαση από ένα βιολογικό μοντέλο σε ένα τεχνητό ψηφιακό μοντέλο. Οι ιδέες της αυτο-οργάνωσης και στιγμεργείας, που προέρχονται από το βιολογικό μοντέλο και την παρατήρηση των πραγματικών μυρμηγκιών, έχουν σημαντικό ρόλο στη συμπεριφορά των τεχνητών αλγορίθμων.

Η στιγμεργεία όπως ορίστηκε στο κεφάλαιο 3, είναι η κυρίαρχη ιδέα πίσω από τη δόμηση των αλγορίθμων μυρμηγκιών, καθώς ο ορισμός στο τεχνητό μοντέλο των λεγόμενων στιγμεργετικών μεταβλητών, δηλαδή των μεταβλητών που περιέχουν τις πληροφορίες που χρησιμοποιούν τα μυρμήγκια για να επικοινωνήσουν έμμεσα μεταξύ τους, είναι το κομβικό σημείο για την αλληλεπίδραση των διαφορετικών τμημάτων του αλγορίθμου. Στους αλγορίθμους που περιγράφονται στο κεφάλαιο 4 ο τρόπος ορισμού της στιγμεργετικής πληροφορίας γίνεται με τον ορισμό της φερομόνης και είναι απόλυτος.

Αλλά τα μυρμήγκια δεν είναι τα μοναδικά κοινωνικά έντομα. Μελέτες γίνονται με βάση των τρόπο αναζήτησης τροφής των μελισσών, των σφηκών και τον τρόπο μετακίνησης σμήνους πτηνών.

Οι μέλισσες ακολουθούν μια περίπλοκη συμπεριφορά για την αναζήτηση της τροφής τους καθώς και για την μεταξύ τους επικοινωνία. Κατά την αναζήτηση τροφής ρυθμίζουν την ταχύτητα τους και την απόσταση που διανύουν με μέσο οπτικοποίησης αυτής της πληροφορίας καθώς πτήση σε χαμηλό ύψος και για μικρή απόσταση συνθέτει την ίδια ποσότητα πληροφορίας με πτήση σε μεγάλο και μεγάλη απόσταση στον εγκέφαλο του εντόμου. Η έκβαση της αναζήτησης της τροφής ανακοινώνεται στα υπόλοιπα μελή της αποικίας με τον χορό των μελισσών που απεικονίζει τη την ποσότητα και η τοποθεσία της τροφής.

Μια αποικία από σφήκες έχει αυστηρή κοινωνική δομή και αυτο-οργάνωση. Η αλληλεπίδραση μεταξύ των μελών της αποικίας και του περιβάλλοντος οδηγεί σε μια δυναμική κατανομή των εργασιών, όπως η αναζήτηση τροφής και η επώαση. Επιπρόσθετα διαθέτουν μια ιεραρχική κοινωνική τάξη η οποία σχηματίζεται από την αλληλεπίδραση των μελών ανά ζεύγη. Αυτή η αναδυόμενη κοινωνική τάξη είναι μία ακολουθία από την πλέον κυρίαρχη στη λιγότερο κυρίαρχη σφήκα. Τα πιθανοκρατικά μοντέλα για την αλληλεπίδραση των σφηκών καθώς και οι εφαρμογές τους βρίσκονται σε αρχικά στάδια.

Επίσης το πρότυπο συμπεριφοράς κοπαδιών ψαριών και σμήνους πτηνών έχει εμπνεύσει νέες τεχνικές όπως η βελτιστοποίηση σμήνους σωματιδίων (particle swarm optimization). Είναι στοχαστική μέθοδος που βασίζεται σε πληθυσμό λύσεων που αλληλεπιδρά. Οι πιθανές λύσεις καλούνται σωματίδια και κινούνται μέσα στο χώρο αναζήτησης ακολουθώντας τα τρέχοντα βέλτιστα σωματίδια. Η μέθοδος έχει εφαρμοστεί σε αρκετά προβλήματα στα οποία δείχνει να αποδίδει ικανοποιητικά.

Συνεπώς παρατήρηση και ανάλυση των προτύπων συμπεριφοράς των κοινωνικών εντόμων αλλά και σμήνους βιολογικών οντοτήτων κατά την αναζήτηση τροφής και/ή την μεταξύ τους αλληλεπίδραση μπορεί να οδηγήσει σε νέα μαθηματικά μοντέλα και αλγορίθμους για την επίλυση δύσκολων προβλημάτων συνδυαστικής βελτιστοποίησης που έχουν εφαρμογή σε πραγματικά προβλήματα.

Arnett R.H., American Insects: A handbook of the insects of America north of Mexico. Van Nostrand Rehinhold, New York, 1985

Ascheuer N., Fischetti M. and Grötschel M., (2001) Solving the Asymmetric Travelling Salesman Problem with time windows by branch-and-cut , Journal of Mathematical Programming Vol 90, Iss 3, p.475-506.

Balas E., Martin C., (1989) The prize collecting travelling salesman problem, Networks Volume 19, Issue 6 , Pages 621 – 636 Willey periodicals.

Bautista J., and Pereira J.(2002) Ant algorithms for assembly line balancing. Proceedings of the Third International Workshop on Ant Algorithms. Lecture Notes In Computer Science; Vol. 2463 p 65-75 , Springer Verlag

Beckers, R., Deneubourg, J.L., Goss, S. , (1992), Trails and U-turns in the Selection of a Path by the Ant Lasius niger, Journal of Theoretical Biology, vo--l 159, p.397-415.

Beni, G., Wang, J. (1989) Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30.

Blum, C. and Dorigo, M. (2004). Deception in ant colony optimization. Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004, Proceedings, vol. 3172, p. 118-129. Lecture Notes in Computer Science. Springer.

Blum, C. and Sampels, M. (2002a). Ant colony optimization for FOP shop scheduling: A case study on different pheromone representations. , Proceedings of the 2002 Congress on Evolutionary Computation (CEC 02), p. 1558-1563, Piscataway, NJ, IEEE Press.

Blum, C. and Sampels, M. (2002b). When model bias is stronger than selection pressure. Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN 02), vol. 2439, in Lecture Notes in Computer Science p. 893-902, Springer Verlag.

Blum, C., Roli, A., and Dorigo, M. (2001). HC-ACO: The hyper-cube framework for Ant Colony Optimization. In Proceedings of MIC'2001 – Meta–heuristics International Conference, volume 2, p. 399-403, Porto, Portugal.

Bonabeau, E., Dorigo, M., and Theraulaz, G., (1999). Swarm Intelligence : From natural to artificial systems, Oxford University Press.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (2000). Inspiration for optimization from social insect behaviour. *Nature* , 406:39.42.

Burbidge J.L., (1991) Production Flow Analysis for Planning Group Technology, Journal of Operations Management, 10 , p. 5-27.

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

Colorni A., Dorigo M., Maniezzo V.,(1992) Distributed optimization by ant colonies. In F.J. Varela & P. Bourgine (Eds.), Proceedings of the first european conference on artificial life. p. 134-142, Cambridge, MA , ,MIT Press.

Cook S. A. (1971) "The complexity of theorem proving procedures". Proceedings, Third Annual ACM Symposium on the Theory of Computing, ACM, New York, p. 151-158

Costa, D. and Hertz, A. (1997). Ants can colour graphs. Journal of the Operational Research Society, issue 48, p. 295-305.

Dammeyer F, Vob S (1993), Dynamic tabu list management using the reverse elimination method, Annals of Operation Research, 41, 31-46

Dantzig G. B. and Ramser R.H. (1959) "The Truck Dispatching Problem". Management Science 6, 80–91.

Dantzig, G.B., Fulkerson, D.R., Johnson, S., (1954) Solution of a large-scale Traveling Salesman Problem. Operations Research, 2, 393–410

Deneubourg, J.-L., Aron, S. Goss, S. Pasteels, J.M., (1990). The Self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behaviour, 3, 159-168.

Dilworth J.B., Operation Management, 2nd Edition, McGraw Hill, 1996

Dorigo M., Bonabeau E., Theraulaz G., Ant algorithms and stigmergy, Future Generation Computer Systems 16 (2000) 851-871

Dorigo M., DiCaro G. Gambardella J.L., (1999), Ant algorithm for discrete optimization, Artificial Life 5 (2), p 137-172

Dorigo M., Maniezzo V., Colorni A., (1991a) Positive feedback as a search strategy.Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo M., Maniezzo V., Colorni A., (1991b) The Ant System: An Autocatalytic Optimizing Process, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo M., Maniezzo V., Colorni A., (1996) Ant System: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics. Part B, 26 (1) p. 29–41.

Dorigo, M. and Di Caro, G. (1999a). Ant Colony Optimization: A new meta-heuristic. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A., editors, Proceedings of the Congress on Evolutionary Computation, volume 2, pages 1470.1477, May_ower Hotel, Washington D.C., USA. IEEE Press.

Dorigo, M. and Stutzle, T. (2001). An experimental study of the simple ant colony optimization algorithm. In Proceedings of the, 2001 WSES International Conference on Evolutionary Computalution Computation. WSES-Press International.

Dorigo, M., Gambardella, L.M., (1997) Ant Colony System: A cooperative learning approach to the travelling salesman problem. IEEE Transactions on Evolutionary Computation, 1 (1), p. 53-66

Dorigo, M., Stützle, T., (2004). Ant Colony Optimization, MIT Press, Cambridge, Massachusetts

Dowsland K.A., Off-the-Peg or Made-to-Measure? Timetabling and Scheduling with SA and TS, in Practice and Theory of Automated Timetabling II, Second International Conference, PATAT'97, Toronto, Canada, August 20-22, 1997, Selected Papers. Lecture Notes in Computer Science 1408 Springer 1998 p: 37-52.

European Cooperation in the Field of Scientific and Technical Research (COST). Cost 339 small containers. Final report of the Action, Energy and Transport Directorate General, European Commission, 2000.

Fainekos G., (2001), The ACO Metaheuristic: Application in discrete and continous problems., Diploma dissertation, Department of Mechanical Engineering,National Technical University of Athens.

Fischetti M., Toth P. (1988) An additive approach for the optimal solution of the prize collecting traveling salesman problem. In Vehicle Routing Methods ad Studies. , North Holland p 319-343.

Gambardella L. M. and Dorigo M. (1997). HAS-SOP: Hybrid ant system for the sequential ordering problem. Technical Report IDSIA-11-97, IDSIA, Lugano, Switzerland.

Gambardella L.M., Dorigo M., (1995) Ant-Q: A reinforcement learning approach to the travelling salesman problem. Proceedings of the Twelfth International Conference on Machine Learning , p. 252-260 , Morgan-Kaufman

Gambardella, L. M. and Dorigo, M. (1996). Solving Symmetric and Asymmetric TSPs by Ant Colonies. p 622-627, In ICEC96, Proceedings of the IEEE Conference on Evolutionary Computation, Japan.

Glover F. (1986) Future paths for integer programming and links to artificial intelligence, Computers and Operation Research, 13, 533-549.

Glover, F., & Laguna, M. , Tabu Search. Boston.Kluwer Academic Publishers 1997.

Goldberg D.E.,GENETIC ALGORITHMS in Search, Optimization and Machine Learning, Addison Wesley Publishing Company, Inc., 1989

Goss, S., Aron S., Deneubourg, J.L., Pasteels, J.M., (1989). Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 76, p. 579-581.

Grassé, P. P., (1959). La Reconstruction du nid et les coordinations interindividuelles chez, Bellicositermes Natalensis et Cubitermes sp. La théorie de la stigmergie: Essai d' interprétation du comportement des termites constructeurs. Insectes-Sociaux., 6, 41-80.

Guntsch M, Middendorf M, (2002 b) A population based approach for ACO, in Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim, vol. 2279, p. 71-80, Springer.

Guntsch M, Middendorf M, (2002 a) Applying Population Based ACO to Dynamic Optimization Problems, in Ant Algorithms, Third International Workshop, ANTS 2002, Proceedings, p. 111-122, vol. 2463 Lecture Notes in Computer Science, Springer-Verlag.

Herault L. (2000) Rescaled Simulated Annealing—Accelerating Convergence of Simulated Annealing by Rescaling the States Energies, Journal of Heuristics, Vol 6, Iss 2 , p. 215-252.

Holland, J.H., (1975), Adaptation in Natural and Artificial Systems, Journal of the ACM,3, 297.

Jaillet P., (1985) The Probabilistic Traveling Salesman Problems. Technical Report Center MIT Cambirdge Mass.

Jeanne, R. L., (1986). The Evolution of the Organization of Work in Social Insects, Monit. Zool. Ital., 20, 119-133.

Karp, R. M. (1972). Reducibility among combinatorial problems. In Complexity of Computer Computations, σελ. 85.103. Plenum Press.

Keinprasit, R. and Chongstitvatana, P. (2004). High-level synthesis by dynamic ant. International Journal of Intelligent Systems, vol 19 issue 1-2,p.25.38.

Kirkpatrick S, Gelatt C Jr., and Vecchi M., (1983), Optimization by simulated annealing, Science Vol 220, No 4598, p 457-464.

Koopmans TC, Beckman M, (1957) Assignment problems and the location of economic activities, Econometrica, Vol. 25, No. 1 , pp. 53-76

Labordere H. (1969) The record balancing problem: A dynamic programming solution of a generalized traveling salesman problem, RIBO B 2, 43-49.

Laporte, G.(1992) The vehicle routing problem: An overview of exact and approximate algorithms, Journal European Journal of Operational Research. Vol 59 Issue 3 : 345-358

Lenstra J. K., Rinnooy Kan A.H.G.,(1981) "Complexity of vehicle routing and scheduling problems", Networks, vol. 11,issue 2, pp. 221-227. 1981

Liu ,J.S. (2001). Monte Carlo Strategies on Scientific Computing, New York, Springer-Verlag

Maniezzo V. and Colorni A. (1999). The ant system applied to the quadratic assignment problem. IEEE Transactions on Knowledge and Data Engineering, vol 11(5), p769-778.

Metropolis, N. Rosenbluth, A. Rosenbluth, M. Teller, A. Teller, E. (1953), Equation of state calculations by fast computing machines. Journal of Chemical Physics, Vol 21, pp. 1087-1092.

Metropolis, N. Rosenbluth, A. Rosenbluth, M. Teller, A. Teller, E. (1953),Equation of state calculations by fast computing machines. Journal of Chemical Physics, Vol 21, pp. 1087-1092.

Miller C.E., Tucker A.W., Zemlin R.A.,(1960) Integer programming formulation of the traveling salesman problem, Journal of Association Computing Machinery, 7:4 p 326-329.

Monmarché N., Venturini G., Slimane M., (2000), On how Pachycondyla apicalis ants suggest a new search algorithm, Future Generation Computer Systems 16, p 937–946.

Nahar S., Sahni S., Shragowitz E., (1986), Simulated Annealing and Combinatorial Optimization, in "Proceedings of the 23rd Design Automation Conference", pp 293-299

Nicolis G., Prigogine I., Self-organization in Non-Equilibrium Systems, Willey & Sons, 1977

Papadimitiou C. , Steiglitz K. (1982) Combinatorial Optimization: Algorithms and Complexity, Prentice Hall ,New Jersey.

Resende, M. and Ribeiro, C. (2003). Greedy randomized adaptive search procedures. Glover, F. and Kochenberger, G. (eds), Handbook of Metaheuristics, p219.249. Kluwer Academic Publishers.

S.Lin and B.Kernigan, (1973), An effective heuristic for the traveling salesman problem, Operations Research, Vol 21, pp 498-516.

Shani S and Gonzalez T (1976). P-complete approximation problems. Journal of the ACM **23**: 555–565.

Shmygelska, A., Hern´andez, R. A., and Hoos, H. H. (2002). An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. Proceedings of the Third International Workshop, ANTS 2002, Proceedings in Lecture Notes in Computer Science (LNCS) series, Vol. 2463, Springer-Verlag.

Silva C. A., Runkler T. A., da Costa Sousa J. M., and Palm R. (2002) Ant Colonies as Logistic Processes Optimizers, Proceedings of the Third International Workshop on Ant Algorithms. Lecture Notes In Computer Science; Vol. 2463 p 76 - 87, Springer Verlag

Socha, K., Sampels, M., and Manfrin, M. (2003). Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art. In Proceedings ofEvoCOP 2003 – 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization, volume 2611 of Lecture Notes in Computer Science, p. 334.345. Springer-Verlag.

Solimanpur M., Vrat P., Shankar R., (2004). Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing, European Journal of Operational Research 157 :592-606

Solomon M. M., (1995) "Algorithms for the Vehicle Routing Problem with Time Windows". Transportation Science, 29(2), pp. 156-166.

Stutzle T,(1998), An ant approach to the flow shop problem, In Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing, p.1560--1564, Verlag Mainz, Aachen, Germany

Stutzle T. & Hoos H.H., (2000) MAX-MIN Ant System. Future Generation of Computer Systems, vol 16 issue 8, p. 889-914

Stutzle, T. (1997). MAX-MIN Ant System for Quadratic Assignment Problems. Technical Report AIDA.97.4, FG Intellektik, TU Darmstadt, Germany.

Stutzle, T. and Dorigo, M. (2002). A short convergence proof for a class of aco algorithms. In IEEE Transactions on Evolutionary Computation, vol 6,issue 4 p 358.365.

Stutzle, T. and Hoos, H. (1997). The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In Back T., Michalewizc Z, Yao X. (Eds.), Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC 97). p. 309-314, IEEE Press

Toth P., Vigo D.(2001): "The Vehicle Routing Problem". Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.

van Laarhoven, P. J. M. (1987). Simulated Annealing: theory and applications, Kluwer Academic Publishers, Dordrecht.

Vlachos A.(2005), Metaheuristic Algorithms and Combinatorial Optimization Problems, PhD, Department of Informatics, University of Piraeus.

Watkins,C, (1989) Learning from Delayed Rewards,Thesis,University of Cambidge,England

Won, You-Dong, (1997) A Linear Programming Approach to Linear Machine Layout Problem, The Journal of the Industrial Mathematics Society, 47, , Part2.