



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ**

**ΓΙΑΚΟΥΜΙΔΑΚΗΣ ΑΝΔΡΕΑΣ**

**Μ.Δ.Ε.:ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ**

**A.M:2014019049**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ**

Θέμα Διατριβής: Βελτιστοποίηση ενός προβλήματος δρομολόγησης οχημάτων (VRP) με χρήση αλγορίθμου Αποικίας Μυρμηγκιών (ACO).

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΜΑΡΙΝΑΚΗΣ ΙΩΑΝΝΗΣ**

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>Εισαγωγή και ευχαριστίες.....</b>	<b>4</b>
<b>Κεφάλαιο 1:LOGISTICS-Έννοια-Εφαρμογές.....</b>	<b>5</b>
1.1 Ορισμός Logistics .....	5
1.2 Τομείς Logistics .....	6
1.3 Logistics Επιχειρήσεων (Business Logistics).....	8
1.3.1 Κόμβοι ενός δικτύου διανομής.....	9
1.3.2 Μεταφορά .....	10
1.4 Εξυπηρέτηση Εφοδιαστικής εκ τρίτων ή Third Party Logistics(3PL).....	10
1.4.1 Ο «Σταθερός» ή «Τυποποιημένος» προμηθευτής 3PL .....	11
1.4.2 Ο «Υπεύθυνος για την Ανάπτυξη Υπηρεσιών» προμηθευτής 3PL.....	12
1.4.3 Ο «Εναρμονιστής» προμηθευτής 3PL .....	12
1.4.4 Ο «Υπεύθυνος για την Ανάπτυξη Πελατών» προμηθευτής 3PL .....	12
<b>Κεφάλαιο 2:Το Πρόβλημα Δρομολόγησης Οχημάτων(VRP). ....</b>	<b>13</b>
2.1 Εισαγωγή.....	13
2.2 Το Πρακτικό Πρόβλημα - Περιορισμοί .....	14
2.3 Στόχοι Δρομολόγησης.....	18
2.4 Το Πρόβλημα του Πλανόδιου Πωλητή .....	20
2.5 Το Πρόβλημα Δρομολόγησης Στόλου Οχημάτων.....	21
2.6 Κατηγορίες προβλημάτων δρομολόγησης οχημάτων(VRP).....	22
<b>Κεφάλαιο 3:Αλγόριθμοι επίλυσης προβλημάτων διανομής.....</b>	<b>28</b>
3.1 Ακριβείς αλγόριθμοι .....	28
3.1.1 Αλγόριθμος <i>Branch and Bound'</i> .....	28
3.2 Ευρετικοί αλγόριθμοι (heuristics) .....	30
3.2.1 Ο αλγόριθμος του πλησιέστερου γείτονα .....	31
3.2.2 Αλγόριθμοι <i>Relocate</i> .....	32
3.2.3 Αλγόριθμος <i>2-opt</i> .....	33
3.2.4 Αλγόριθμος <i>1-1 Exchange</i> .....	34
3.3 Μεθευρετικοί αλγόριθμοι (Metaheuristics).....	35
3.3.1 Προσομοιωμένη Ανόπτηση( <i>Simulated Annealing</i> ) .....	35
3.3.2 Περιορισμένη Αναζήτηση( <i>Tabu Search</i> ) .....	38
3.3.3 Γενετικοί Αλγόριθμοι( <i>Genetic Algorithms</i> ) .....	42
3.3.4 Αλγόριθμος Βελτιστοποίησης Αποικίας Μυρμηγκιών( <i>ACO</i> ) .....	46
<b>Κεφάλαιο 4: Περιγραφή Αλγορίθμου .....</b>	<b>53</b>
4.1: Διάβασμα μεταβλητών, αρχικοποιήσεις.....	53
4.2:Παραγωγή αρχικού αποτελέσματος με αλγόριθμο πλησιέστερου γείτονα.....	55

4.3: Αρχικοποίηση μεταβλητών για αλγόριθμο ACO και ο αλγόριθμος επιλογής Ρουλέτας(Roulette Wheel Selection). ....	57
4.4: Διερργασίες μετά την ολοκλήρωση διαδρομών μυρμηγκιών. ....	61
4.5: Διερργασίες μετά την ολοκλήρωση του εξωτερικού βρόγχου και επεξεργασία τελικής διαδρομής.....	63
<b>Κεφάλαιο 5: Αποτελέσματα και συμπεράσματα.....</b>	<b>65</b>
<b>Βιβλιογραφία .....</b>	<b>81</b>

# Εισαγωγή και ευχαριστίες

Η παραπάνω διατριβή όπως αναφέρθηκε ασχολείται με μια αλγοριθμική προσέγγιση σε προβλήματα βελτιστοποίησης διαδρομών που προκύπτουν κατά την εκτέλεση εργασιών Logistics. Ο όρος Logistics όπως θα δούμε και παρακάτω δεν έχει να κάνει με λογιστικές εργασίες, όπως συχνά μπερδεύεται κυρίως από Ελληνόφωνους αναγνώστες, αλλά με την διαδικασία που ακολουθείται για την ομαλή λειτουργία της εφοδιαστικής αλυσίδας σε ένα πλήθος εφαρμογών.

Η πολυπλοκότητα διαφέρει από εφαρμογή σε εφαρμογή και δεν είναι πάντα δυνατό ή προφανές το πώς θα βρεθεί μια απόλυτα βέλτιστη λύση. Οι περιορισμοί κάθε είδους αλλά και η δυναμικότητα πολλών προβλημάτων χρήζει αντιμετώπισης με χρήση αλγορίθμων που προσεγγίζουν ή φτάνουν μια λύση που μας παρέχει το βέλτιστο αποτέλεσμα και συνεπώς ελάχιστο κόστος στις πιο πολλές περιπτώσεις.

Για τον παραπάνω σκοπό αναπτύχθηκε κώδικας σε Matlab, στον οποίο γίνονται εισαγωγή τα δεδομένα του εκάστοτε προβλήματος (Θέση- πλήθος κόμβων, διαθεσιμότητα, μέγιστη διάρκεια διαδρομής κάθε οχήματος κτλ.). Ο λόγος φυσικά για πρόβλημα δρομολόγησης οχημάτων VRP (Vehicle Routing Problem).

Στο συγκεκριμένο αρχείο η αρχή γίνεται με την επεξήγηση των Logistics σαν έννοια, σκοπό και σύγχρονα προβλήματα που αντιμετωπίζει. Στη συνέχεια παρουσιάζεται σαν έννοια το πρόβλημα δρομολόγησης οχημάτων καθώς και αναφορές αλγορίθμων για την επίλυση του με έμφαση στον αλγόριθμο αποικίας μυρμηγκιών ACO (Ant Colony Optimization). Η πορεία των προσπαθειών βέλτιστης επίλυσης του κατά τη διάρκεια πολλών ετών είναι πολύ σημαντική για την βαθύτερη κατανόηση του προβλήματος πριν καταλήξουμε στον τρόπο που προσεγγίστηκε σε αυτήν στην διατριβή και περιγράφει η λειτουργία του αλγορίθμου. Εν συνεχεία θα γίνει παρουσίαση αποτελεσμάτων και συμπερασμάτων με την απαραίτητη για την σύνταξη του συγκεκριμένου εγγράφου βιβλιογραφίας.

Τέλος θα ήθελα να ευχαριστήσω όσους μου συμπαραστάθηκαν κατά την εκπόνηση της Μεταπτυχιακής διατριβής με οποιονδήποτε τρόπο και ιδιαιτέρως τον Επιβλέποντα καθηγητή **κ. Ιωάννη Μαρινάκη** αλλά και την **κα. Μαγδαληνή Μαρινάκη**, χωρίς την πολύτιμη βοήθεια των οποίων δεν θα ήταν δυνατή η ολοκλήρωση της.

# Κεφάλαιο 1: LOGISTICS-Έννοια-Εφαρμογές

## 1.1 Ορισμός Logistics

Τα Logistics είναι γενικά η λεπτομερής οργάνωση και υλοποίηση μιας σύνθετης ενέργειας. Από επιχειρηματική άποψη , logistics είναι η διαχείριση της ροής των πραγμάτων μεταξύ του σημείου προέλευσης και του σημείου της κατανάλωσης, προκειμένου να ικανοποιούνται οι απαιτήσεις των μεμονωμένων πελατών ή εταιρειών. Οι πόροι διαχείρισης των logistics μπορεί να περιλαμβάνουν φυσικά στοιχεία, όπως τα τρόφιμα, τα υλικά, τα ζώα, εξοπλισμό ,ακόμα και υγρά καθώς και στοιχεία, όπως είναι ο χρόνος και γενικά πληροφορίες. Τα logistics των φυσικών αντικειμένων συνήθως περιλαμβάνουν την ενσωμάτωση της ροής πληροφοριών, το χειρισμό υλικών, την παραγωγή, τη συσκευασία, την απογραφή, μεταφορά και αποθήκευση.

Στον Στρατιωτικό τομέα, η εφοδιαστική ασχολείται με τη διατήρηση των γραμμών ανεφοδιασμού του στρατού, ταυτοχρόνως και με την φθορά της αντίστοιχης του εχθρού, δεδομένου ότι μια ένοπλη δύναμη, χωρίς πόρους και μεταφορά αναγκαίων προμηθειών είναι ανυπεράσπιστα.

Σε οποιοδήποτε τομέα και αν εφαρμόζονται τα Logistics , προκύπτει η ανάγκη κεντρικού ελέγχου και συνεπώς η διαχείριση τους, υπεύθυνη για το σχεδιασμό, υλοποίηση και έλεγχο της αποδοτικότητας και αποτελεσματικότητας αποθήκευσης των αγαθών, των υπηρεσιών και των σχετικών πληροφοριών μεταξύ του σημείου προέλευσης και του σημείου της κατανάλωσης, προκειμένου να ανταποκριθεί στις απαιτήσεις του πελάτη. Η διαδικασία αυτή , ιδιαίτερα σε επίπεδο μεγάλων εταιριών είναι αρκετά πολύπλοκη και όπως είναι φυσικό ,βασικό στοιχείο που ζητείται είναι η ελαχιστοποίηση της χρήσης των πόρων σε όλους τους τομείς της εφοδιαστικής.

## 1.2 Τομείς Logistics

**Logistics εισερχόμενης κίνησης** είναι μια από τις κύριες διαδικασίες των logistics, με επίκεντρο την αγορά και την προετοιμασία της εισερχόμενης κίνησης των υλικών, εξαρτημάτων, και / ή τελικών αποθεμάτων από τους προμηθευτές σε αποθήκες ή καταστήματα λιανικής πώλησης.

**Logistics εξερχόμενης κίνησης** είναι η διαδικασία που σχετίζεται με την αποθήκευση και τη διακίνηση του τελικού προϊόντος καθώς και της ροής πληροφοριών από το τέλος της γραμμής παραγωγής προς τον τελικό χρήστη.

Λαμβάνοντας υπόψη τις υπηρεσίες που εκτελούνται, οι κύριοι τομείς των logistics μπορούν να αναλυθούν ως εξής:

- Logistics Προμηθειών
- Logistics Διανομών
- Logistics μετά πωλήσεως
- Logistics απόθεσης απορριμμάτων και αποβλήτων
- Logistics επιστροφών
- “Πράσινα” Logistics
- RAM logistics
- Logistics διαχείρισης κρίσεων
- Logistics Παραγωγής
- Κατασκευαστικά Logistics
- Ψηφιακά Logistics

Ακολουθεί συνοπτική περιγραφή των πιο σημαντικών εξ’ αυτών:

**Logistics Προμηθειών:** Περιλαμβάνει διαδικασίες όπως Έρευνα αγοράς, σχεδιασμό και πρόβλεψη απαιτήσεων, αποφάσεις αγοραπωλησιών, διαχείριση προμηθειών-παραγγελιών. Οι στόχοι στον τομέα αυτό συνοψίζονται ως: μεγιστοποίηση της αποτελεσματικότητας με επίκεντρο τις βασικές ικανότητες, την εξωτερική ανάθεση, διατηρώντας παράλληλα την αυτονομία της εταιρείας ή την ελαχιστοποίηση του κόστους προμήθειας με μεγιστοποίηση της ασφάλειας στο πλαίσιο της διαδικασίας της προσφοράς.

**Logistics Διανομών:** Εδώ οι βασικές διεργασίες είναι αυτές που κρίνονται απαραίτητες για την παράδοση του τελικού προϊόντος στον πελάτη. Η επεξεργασία παραγγελιών, αποθήκευση και μεταφορά είναι απαραίτητες για αυτό το κομμάτι καθώς ο χρόνος, η ποσότητα του προϊόντος αλλά και η

τοποθεσία κατά την διάρκεια της παράγωγής, διαφέρουν από τις αντίστοιχές τους έννοιες της κατανάλωσης.

**Logistics απόθεσης απορριμμάτων και αποβλήτων:** Σε αυτό το κομμάτι, η ελαχιστοποίηση του κόστους ταυτόχρονα με την ενίσχυση διαδικασιών που σχετίζονται με την απόθεση απορριμμάτων-αποβλήτων τα οποία παράγονται κατά την λειτουργία μιας επιχείρησης, είναι το κύριο αντικείμενο.

**Logistics επιστροφών:** δηλώνονται όλες οι εργασίες που σχετίζονται με την επαναχρησιμοποίηση των προϊόντων και υλικών. Η διαδικασία της αντίστροφης εφοδιαστικής όπως μεταφράζεται , περιλαμβάνει τη διαχείριση και την πώληση των πλεονασμάτων, καθώς και τα προϊόντα που επιστρέφονται στους προμηθευτές από τους αγοραστές. Είναι η διαδικασία του σχεδιασμού, εφαρμογής και ελέγχου της απόδοσης ροής των πρώτων υλών, του αποθέματος και ανάκτησης-υπολογισμού αξίας προϊόντων που επιστρέφονται από τον τελικό προορισμό, πιθανή καταστροφή προϊόντων αναλόγως την φύση και την κατάσταση των αγαθών και επαναπροώθηση στην αγορά. Χαρακτηριστικό παράδειγμα είναι η επαναπροώθηση γαλακτοκομικών προϊόντων μετά την λήξη της παστερίωσης.

**“Πράσινα” Logistics** τα οποία περιγράφουν τις προσπάθειες εκτίμησης αλλά και ελαχιστοποίησης οικολογικού αντίκτυπου κατά την εκτέλεση διεργασιών logistics. Εκτός από κατάλληλη επιλογή οχημάτων (χωρητικότητα , κατανάλωση κτλ), εδώ τίθεται και το θέμα ελαχιστοποίησης διαδρομής που αποτελεί και μέρος αυτής της διατριβής.

**RAM Logistics** τα οποία συνδυάζουν **Logistics επιχειρήσεων** (θα ασχοληθούμε περεταίρω) και **Στρατιωτικά Logistics**. Εδώ έχουμε χρήση πολύπλοκων και αξιόπιστων συστημάτων τεχνολογίας αιχμής, λαμβάνοντας υπόψη και την αύξηση της αξίας των αντικειμένων (μεταφορά οπλικών συστημάτων , πυραύλων , φορτίων με προϊόντα τεχνολογίας τεράστιας εμπορικής αξίας). Τα αρχικά RAM εξηγούνται ως Reliability(αξιοπιστία), Availability(διαθεσιμότητα) and Maintainability(συντηρησιμότητα).

**Logistics Διαχείρισης Κρίσεων** Είναι ένας όρος που χρησιμοποιείται στην εφοδιαστική αλυσίδα και την βιομηχανία αναφερόμενος σε συγκεκριμένες χρονικές στιγμές στις οποίες κρίνεται απαραίτητη η ταχύτατη μεταφορά αγαθών διαφόρων αγαθών ή αντικειμένων σε προορισμό όπου έχει προκύψει κατάσταση εκτάκτου ανάγκης .Ο λόγος που σαν υπηρεσία αναφέρονται ξεχωριστά είναι προβλήματα τα οποία παρουσιάζονται ενδεχομένως (απαγορευτικά απόπλου ή πτήσεων, καθυστέρηση παραγωγής, καταστροφή τηλεπικοινωνιακού δικτύου σε περιπτώσεις πολέμου κτλ). Σε αυτές τις περιπτώσεις χρειάζονται να ληφθούν συγκεκριμένα μέτρα για να μην επηρεαστεί σε σοβαρό βαθμό η εφοδιαστική διαδικασία.



Εικόνα 1 Forklift στο Πακιστάν φορτώνει υλικά ανθρωπιστικής βοήθειας κατά την περίοδο της καταστροφικής πλημμύρας του 2010.

**Logistics Παραγωγής**, τα οποία περιγράφουν τις διαδικασίες που ακολουθούνται σε συστήματα προστιθέμενης αξίας( Ορυχεία ή εργοστάσια). Σκοπός είναι να διασφαλισθεί ότι κάθε μηχανήμα ή σταθμός εργασίας λαμβάνει την σωστή ποιότητα, ποσότητα και φυσικά το σωστό προϊόν στις προβλεπόμενες χρονικές στιγμές. Κυριότερα αντικείμενα προς επεξεργασία είναι η παραγωγή, ο έλεγχος ποιότητας, μεταφορά, αποθήκευση και προμήθεια. Εφαρμογή υπάρχει κυρίως σε μονάδες παραγωγής-εργοστάσια όπου οι διαδικασίες συνεχώς αλλάζουν είτε λόγω αύξησης-μείωσης της ζήτησης είτε λόγω αντικατάστασης μηχανημάτων. Ο συγκεκριμένος τομέας logistics παρέχει τον τρόπο ώστε να υπάρχει ανταπόκριση στις παραγγελίες πελατών με όσο το δυνατό αποδοτικότερο από οικονομικής άποψης τρόπο. Νευραλγικό κομμάτι είναι η μείωση παρτίδας όποτε κριθεί απαραίτητο. Σε πολλές βιομηχανίες όπως αυτή της κινητής τηλεφωνίας ο στόχος είναι ακόμα και η ζήτηση ενός πελάτη να εξυπηρετείται αξιόπιστα και αποδοτικά. Η διαδικασία παρακολουθήσεως και εντοπισμού (Track and tracing), η οποία είναι σημαντικό μέρος των Logistics παραγωγής, εξαιτίας θεμάτων όπως ασφάλεια και αξιοπιστία προϊόντος, αποκτά βαρύνουσα σημασία κυρίως στον τομέα της αυτοκινητοβιομηχανίας και της φαρμακοβιομηχανίας.

### 1.3 Logistics Επιχειρήσεων (Business Logistics)

Ένας ορισμός των επιχειρηματικών Logistics αναφέρεται ως «ύπαρξη του σωστού προϊόντος στην σωστή ποσότητα και ποιότητα , την σωστή στιγμή και στο σωστό μέρος με την κατάλληλη τιμή και προβλεπόμενη κατάσταση στους σωστούς πελάτες».



Ο όρος αυτός έχει εξελιχθεί από την δεκαετία του '60 εξαιτίας της αυξανόμενης πολυπλοκότητας των αντιστοιχών εταιριών με τα υλικά αλλά και την αποστολή προϊόντων να γίνονται πλέον σε παγκόσμια κλίμακα και την ανάγκη πλέον επαγγελματιών του κλάδου με τον διεθνή όρο "Supply Chain Logisticians".

Όπως αναφέρθηκε και σε προηγούμενο υποκεφάλαιο, στον τομέα των επιχειρήσεων του κλάδου τα Logistics μπορεί να επικεντρώνονται εσωτερικά (in-bound logistics) ή εξωτερικά (out-bound logistics) καλύπτοντας την ροή και αποθήκευση υλικών από το σημείο προέλευσης προς το σημείο κατανάλωσης, (supply-chain management). Οι κύριες λειτουργίες που λαμβάνουν χώρα είναι διαχείριση αποθέματος, αγορά, μεταφορά, αποθήκευση, καθώς και η οργάνωση των αναφερθέντων δραστηριοτήτων.

Υπάρχουν 2 θεμελιώδεις και διακριτές μορφές Logistics: Η μία βελτιστοποιεί μια σταθερή ροή προϊόντων μέσω ενός δικτύου συνδέσμων μεταφοράς (transport links) και κόμβων αποθήκευσης, ενώ η άλλη έχει να κάνει με διαχείριση πόρων (αποθήκες, οχήματα) προς επίτευξη του τελικού στόχου.

### 1.3.1 Κόμβοι ενός δικτύου διανομής

Οι κόμβοι ενός δικτύου διανομής περιλαμβάνουν:

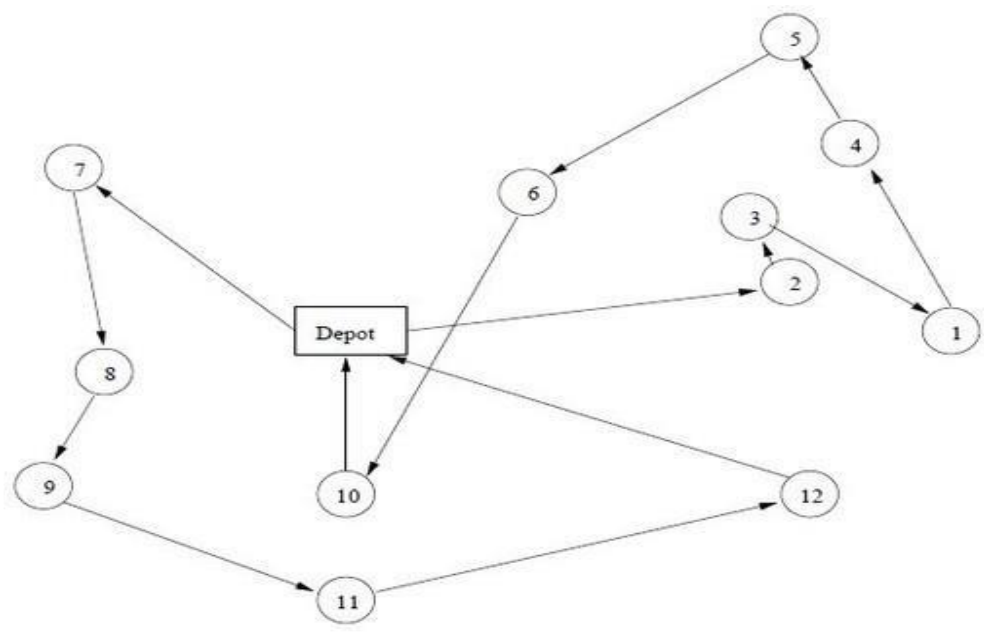
- Εργοστάσια όπου τα προϊόντα παράγονται και/ή συναρμολογούνται
- Μια αποθήκη ή περισσότερες για αποθήκευση εμπορευμάτων.
- Κέντρα διανομής (Distribution centers) για επεξεργασία παραγγελιών και εκπλήρωσή τους (order processing and order fulfillment) και επίσης για παραλαβή αντικειμένων που για διάφορους λόγους επιστρέφονται από τους πελάτες.
- Σημεία διέλευσης για cross docking activities, όπου επανατοποθετούνται και επαναπροσδιορίζονται τα προς αποστολή προϊόντα αναλόγως τους πελάτες. Ουσιαστικά ξεφόρτωμα και φόρτωμα πάλι.
- Καταστήματα πώλησης των προϊόντων (λιανικής ή χονδρικής) (Mom and Pop) Σούπερ μάρκετ κτλ.

Ενδεχομένως να υπάρχουν και μεμονωμένα άτομα που διαχειρίζονται τις εργασίες μεταξύ των κόμβων (sales agents ή brokers). Θα αναφερθούμε αναλυτικότερα σε αυτό στο τέλος του παρόντος κεφαλαίου.

### 1.3.2 Μεταφορά

Το εμπόρευμα προς διανομή, μπορεί να μεταφερθεί με μια ποικιλία από μέσα και διαχωρίζεται σε διαφορετικές κατηγορίες (shipment categories). Φορτία ενός συγκεκριμένου είδους, κατανεμημένα σε παλέτες συνήθως μεταφέρονται με μέσα όπως Containers. Κατά την μετακίνηση φορτίου, τυπικοί περιορισμοί είναι το μέγιστο βάρος και όγκος.

Επίσης κατά την μετακίνηση περιλαμβάνονται μέσα μεταφοράς όπως: τρέινα, οχήματα δρόμου, καράβια, αεροπορικές εταιρείες, Courier κτλ. Εμπορεύματα τα οποία μεταφέρονται διεθνώς πρέπει να συμφωνούν με τα standards του Incoterms σύμφωνα με τους κανόνες που επιτηρούνται από τον Διεθνή Οργανισμό Εμπορίου.



Εικόνα 2 Ενδεικτικό μεταφορικό δίκτυο με κόμβους και ακμές.

### 1.4 Εξυπηρέτηση Εφοδιαστικής εκ τρίτων ή Third Party Logistics(3PL)

Ένας **third-party logistics προμηθευτής (3PLs εταιρίες** σε συντομογραφία, ή μερικές φορές **TPL**) είναι η επιχείρηση που παρέχει μια υπηρεσία σημείων ενιαίας εξυπηρέτησης στους πελάτες της και αφορά εξωτερικές (ή τρίτες) υπηρεσίες μεταφοράς και αποθήκευσης, είτε για ένα μέρος είτε για

ολόκληρη την λειτουργία της εφοδιαστικής τους αλυσίδας. Κύριος στόχος των υπηρεσιών αυτών είναι τα προϊόντα των πελατών τους να καταλήξουν στον τελικό χρήστη στον μικρότερο δυνατό χρόνο και το χαμηλότερο δυνατό κόστος, προσφέροντας εξαιρετικής ποιότητας εξυπηρέτηση και απόλυτη εξειδίκευση ανάλογα με το τι απαιτήσεις και ανάγκες έχουν τα αγαθά που μεταφέρουν.

Οι third-party logistics προμηθευτές ως επί το πλείστον ειδικεύονται στον τομέα της ολοκληρωμένης παροχής υπηρεσιών λειτουργίας, αποθήκευσης και μεταφοράς. Έχουν την δυνατότητα να κλιμακωθούν και να προσαρμοστούν στις ανάγκες του κάθε πελάτη, βασιζόμενοι στις εκάστοτε συνθήκες που επικρατούν στην αγορά, στις απαιτήσεις και στις υπηρεσίες παράδοσης που χρειάζονται τα προϊόντα και τα υλικά τους.

Μια γενική θεώρηση της έννοιας αυτής, είναι ότι αποτελούν μια επιχείρηση, η οποία παρέχει πολλαπλές υπηρεσίες logistics προς χρήση για τους πελάτες της. Κατά προτίμηση, οι υπηρεσίες αυτές είναι ενοποιημένες και διαρθρωμένες μεταξύ τους από τον προμηθευτή. Μεταξύ των υπηρεσιών που προσφέρουν οι **3PLs εταιρείες** περιλαμβάνονται η μετακίνηση, η αποθήκευση, η διαχείριση των αποθεμάτων, η συσκευασία και η αποστολή των εμπορευμάτων στους τελικούς αποδέκτες.

Ουσιαστικά θα λέγαμε με τέτοιου είδους επιχειρήσεις δίνεται η δυνατότητα να χρησιμοποιούνται πόροι συνεργατών, όπως άνθρωποι, υλικοί, τεχνολογικοί κλπ. αποσκοπώντας στην συνεχή βελτίωση της λειτουργίας του κυκλώματος logistics μιας εταιρίας. Με την σύναψη μιας τέτοιας, μακροχρόνιας συνθήκης, συμφωνίας μπορεί να μειωθεί το κόστος της εφοδιαστικής, λόγω της μη χρησιμοποίησης ιδίων κεφαλαίων, ενώ παράλληλα υπάρχει διαθέσιμος ένας εξειδικευμένος και έμπειρος συνεργάτης. Επιπρόσθετα, τέτοιου είδους συνεργασίες πραγματοποιούνται συνήθως είτε όταν μια επιχείρηση προμηθεύει προϊόντα σε αγορές που βρίσκονται μακρύτερα από την βάση των αποθηκών της (μείωση μεταφορικών εξόδων), είτε όταν επιθυμεί να επεκταθεί σε νέες αγορές κυρίως του εξωτερικού. Στην τελευταία περίπτωση οι **3PLs εταιρίες** είναι καλύτεροι γνώστες των νέων αυτών αγορών, με αποτέλεσμα να συμβάλλουν καίρια και καθοριστικά στην εγκαθίδρυση, αποδοχή και ανάπτυξη της εταιρίας.

#### 1.4.1 Ο «Σταθερός» ή «Τυποποιημένος» προμηθευτής 3PL

Αυτή η κατηγορία συνιστά την βασική μορφή των third-party logistics. Εκτελούν δραστηριότητες όπως η συλλογή και η συσκευασία, η αποθήκευση και η διανομή των εμπορευμάτων σε επιχειρήσεις, δηλαδή τις βασικότερες

λειτουργίες της εφοδιαστικής. Για την πλειοψηφία των επιχειρήσεων αυτών η 3PL λειτουργία δεν αποτελεί την κύρια δράση τους.

#### 1.4.2 Ο «Υπεύθυνος για την Ανάπτυξη Υπηρεσιών» προμηθευτής 3PL

Αυτό το είδος 3PL προμηθευτή έχει την δυνατότητα είτε να προσφέρει στους πελάτες του προηγμένες υπηρεσίες προστιθέμενης αξίας, όπως η παρακολούθηση και ο εντοπισμός (tracking and tracing) και ειδικευμένους τρόπους συσκευασίας, είτε να παρέχει ένα πρωτότυπο και μοναδικό σύστημα ασφαλείας. Τα γερά IT (information technology) θεμέλια, η εστίαση σε οικονομικές κλίμακες και το πεδίο εφαρμογής του επιτρέπουν σε αυτού του είδους προμηθευτή 3PL την εκτέλεση αυτών των καθηκόντων αποτελεσματικά και με ακρίβεια.

#### 1.4.3 Ο «Εναρμονιστής» προμηθευτής 3PL

Η μορφή αυτή περιλαμβάνει τους προμηθευτές 3PLs που μπορούν και εναρμονίζονται, δηλαδή προσαρμόζονται στις ανάγκες και τις επιθυμίες των πελατών τους. Αναλαμβάνει, μετά από απαίτηση του πελάτη, στην ουσία τον πλήρη έλεγχο και την διαχείριση των logistics της επιχείρησης. Η 3PL εταιρία βελτιώνει εντυπωσιακά την εφοδιαστική, αλλά σε καμία περίπτωση δεν αναπτύσσει μια νέα υπηρεσία. Η πελατειακή βάση για αυτές τις 3PL επιχειρήσεις είναι κατά κανόνα αρκετά μικρή.

#### 1.4.4 Ο «Υπεύθυνος για την Ανάπτυξη Πελατών» προμηθευτής 3PL

Αυτό είναι το υψηλότερο δυνατό επίπεδο που ένας προμηθευτής 3PL μπορεί να επιτύχει σε σχέση με τις διαδικασίες και τις δραστηριότητες του. Το γεγονός αυτό εμφανίζεται όταν ο 3PL φορέας ενσωματώνεται με τις δραστηριότητες του πελάτη του και αναλαμβάνει εξολοκλήρου την λειτουργία της εφοδιαστικής του. Οι τελευταίοι προμηθευτές 3PL έχουν λίγους πελάτες μεν αλλά εκτελούν μεγάλο αριθμό καθηκόντων, ακόμη και άκρως λεπτομερή γι' αυτούς.

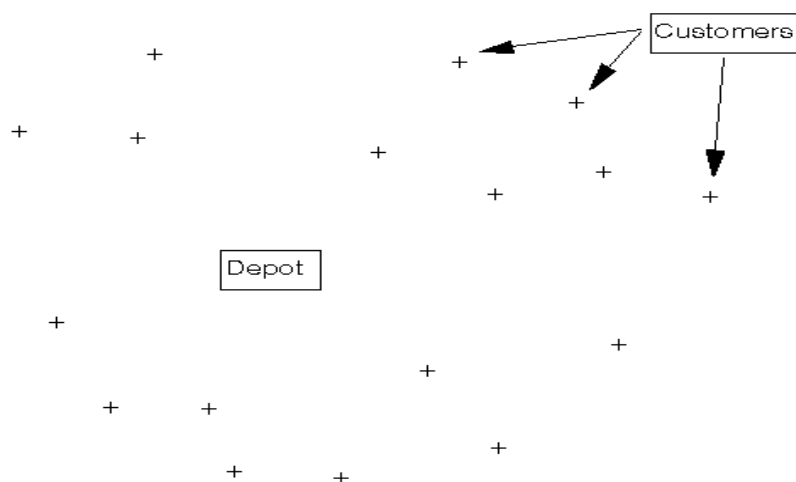
## Κεφάλαιο 2: Το Πρόβλημα Δρομολόγησης Οχημάτων (VRP).

### 2.1 Εισαγωγή

Μια απαραίτητη εργασία, που πρέπει να ολοκληρωθεί στα logistics, είναι η δρομολόγηση οχημάτων. Οι σύγχρονες τεχνολογίες και τα πληροφοριακά συστήματα επιτρέπουν την καλύτερη επικοινωνία με τα οχήματα και την οργάνωση των διαδρομών. Όλη η τεχνολογική πρόοδος οδηγεί σε ακόμα πιο καινοτόμες προσεγγίσεις του προβλήματος και σε ακόμα πιο αποδοτικές λύσεις.

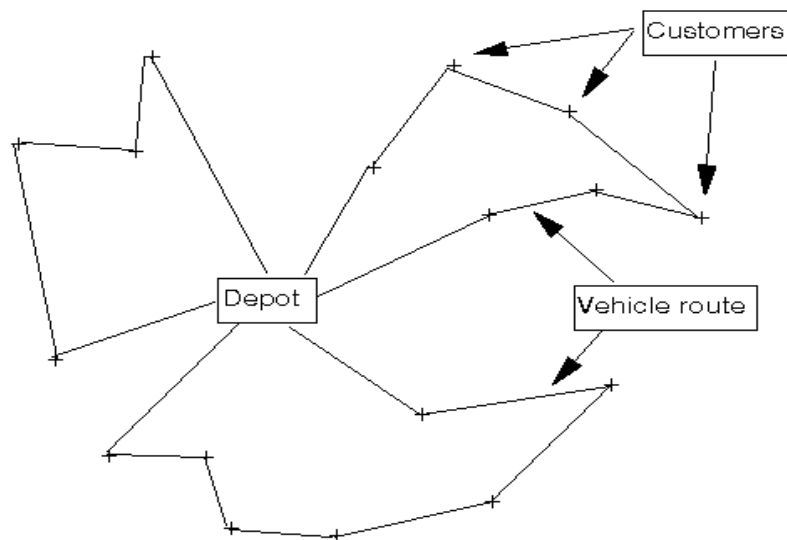
Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem), είναι στην ουσία ένα πρόβλημα συνδυαστικής βελτιστοποίησης. Επιδιώκεται η εξυπηρέτηση ενός αριθμού πελατών με ένα διαθέσιμο στόλο οχημάτων. Στη συνηθισμένη του μορφή έχει στόχο την ελαχιστοποίηση του κόστους παράδοσης στους πελάτες, που καταχωρούν παραγγελίες, των αγαθών που βρίσκονται σε μια κεντρική αποθήκη. Πολλές είναι οι παραλλαγές του προβλήματος, καθώς και οι μέθοδοι επίλυσης που έχουν προταθεί, αφού η εύρεση του ελάχιστου της συνάρτησης κόστους είναι μια σύνθετη διαδικασία.

Υποθέτοντας ένα πρόβλημα δρομολόγησης οχημάτων όπως παρουσιάζεται παρακάτω, όπου μια αποθήκη είναι περιτριγυρισμένη από πελάτες, οι οποίοι προμηθεύονται από εκεί προϊόντα. Η αποθήκη αναλαμβάνει να δημιουργήσει τις διαδρομές που θα ακολουθηθούν από τα οχήματα για τη διανομή των προϊόντων στους πελάτες.



Εικόνα 3 Αποθήκη (Depot) και πελάτες (Customers).

Στο επόμενο σχήμα βλέπουμε τις διαδρομές των οχημάτων από την αποθήκη προς τους πελάτες. Στη συγκεκριμένη περίπτωση, το VRP μετατρέπεται σε ένα πρόβλημα σχεδιασμού διαδρομών οχημάτων γνωστής χωρητικότητας, με στόχο την διανομή των προϊόντων στους πελάτες, σε συγκεκριμένο χρόνο, για να καλύπτεται η ζήτηση.



Εικόνα 4Αποθήκη, πελάτες, διαδρομές οχημάτων.

Οι διαδρομές των οχημάτων σχεδιάζονται έτσι ώστε να ελαχιστοποιείται η συνολική απόσταση που θα διανύσουν τα οχήματα. Η παραπάνω περίπτωση είναι μια κλασσική περίπτωση του προβλήματος. Βεβαίως, το VRP έχει πολλές παραλλαγές και ως πρόβλημα έχει απασχολήσει ευρέως την επιστημονική κοινότητα. Ακολούθως, το VRP θα αναλυθεί σε διάφορες εκφάνσεις του.

## 2.2 Το Πρακτικό Πρόβλημα - Περιορισμοί

Το Πρόβλημα Δρομολόγησης Οχημάτων μπορεί να περιλαμβάνει πολλούς περιορισμούς στις διαδρομές που δύναται να ακολουθήσουν τα οχήματα (π.χ. ώρες εργασίας προσωπικού) και περιορισμούς που έχουν να κάνουν με τους πελάτες. Στην ουσία, πρόσθετοι περιορισμοί και αλλαγές στις απαιτήσεις διαμορφώνουν και τις παραλλαγές του προβλήματος που συναντάμε στην πράξη. Σε μια προσπάθεια να συγκεντρώσουμε περιορισμούς που συναντάμε σε τέτοια προβλήματα [Sbihi, Eglese, 2007, Erdogan, Miller-Hooks, 2011, Μουρκούσης, 2008] έχουμε :

### **1)Περιορισμούς στα σημεία παραγωγής ή τις αποθήκες διανομής**

- Η ύπαρξη περισσότερων του ενός αρχικών σημείων παραγωγής ή διανομής από τα οποία πρέπει να εξυπηρετηθούν οι πελάτες. Σε αυτή την περίπτωση το πρόβλημα γίνεται αρκετά πολυσύνθετο. Σε πολλές περιπτώσεις υπάρχει δυνατότητα μέσα από τη μελέτη να προκύψουν βιώσιμες οικονομικά προτάσεις είτε για τη δημιουργία νέων αποθηκών είτε για τη μείωσή τους.
- Υπάρχει το ενδεχόμενο σε κάθε χρονική στιγμή να μην είναι διαθέσιμα όλα τα σημεία παραγωγής. Ο προγραμματισμός της διανομής δυσκολεύει ακόμα περισσότερο σε τέτοιες περιπτώσεις.
- Η πιθανή διαφοροποίηση ως προς τα προϊόντα, τα οποία είναι διαθέσιμα σε διαφορετικά σημεία παραγωγής ή σε διαφορετικές αποθήκες.

### **2)Περιορισμούς ως προς το στόλο οχημάτων**

- Τα οχήματα έχουν όριο χωρητικότητας μεταφερόμενων εμπορευμάτων. Αυτός ο περιορισμός είναι κομβικός σε ένα VRP, καθώς υπάρχει το ενδεχόμενο να έχουμε οχήματα συγκεκριμένης χωρητικότητας για την εξυπηρέτηση των πελατών. Σε πολλά προβλήματα έχουμε να επιλέξουμε ανάμεσα σε τύπους οχημάτων διαφορετικής χωρητικότητας, αυτόν που είναι ιδανικός για την κάλυψη των αναγκών της εταιρείας. Επιπλέον, μπορεί να απαιτείται η δόμηση ενός ανομοιογενούς στόλου οχημάτων για την αποδοτικότερη λειτουργία της διανομής.
- Το κόστος του κάθε οχήματος είναι ένας καθοριστικός παράγοντας στην εξίσωση. Η εταιρεία οφείλει στους υπολογισμούς της να συνυπολογίσει το κόστος απόκτησης, συντήρησης και κίνησης (καύσιμα) διαφορετικών τύπων οχημάτων. Πολλές φορές όλο αυτό ενσωματώνεται σε ένα σταθερό κόστος για την διανομή, ώστε να έχει σε υπηρεσία ένα όχημα, και σε ένα μεταβλητό κόστος, ανάλογα με τις χιλιομετρικές αποστάσεις που καλύπτονται. Ο τρόπος κοστολόγησης εξαρτάται και από το αν τα οχήματα είναι ιδιόκτητα ή ενοικιαζόμενα.
- Ο χρόνος, που πρέπει να μεσολαβήσει από την αναχώρηση του οχήματος από την αποθήκη έως την επιστροφή του, υπόκεινται σε περιορισμούς είτε νομικούς (π.χ. ώρες απασχόλησης οδηγών, νωπά προϊόντα) είτε άλλους (π.χ. ώρες λειτουργίας εγκαταστάσεων, φόρτος

διελεύσεων στις αποθήκες). Επιπλέον, στόχος σε τέτοια προβλήματα είναι η βέλτιστη αξιοποίηση του εξοπλισμού της εταιρείας.

- Τα διαστήματα στα οποία ένα όχημα παραμένει άεργο. Αυτό οφείλεται σε διάφορους λόγους. Για παράδειγμα, οι οδηγοί έχουν συγκεκριμένο ωράριο εργασίας και τα οχήματα πρέπει να υπόκεινται σε τεχνικούς ελέγχους και συντηρήσεις.
- Τεχνολογικές διαφοροποιήσεις οχημάτων. Υπάρχουν οχήματα κάθε τύπου : διαφορετικά μεγέθη, διαφορετική ιπποδύναμη ή αυτονομία, ψυγεία, βυτία. Έτσι, έχουμε διαφορετικές ταχύτητες κίνησης στο οδικό δίκτυο, διαφορετικούς χρόνους φορτοεκφόρτωσης, διαφορετικές προδιαγραφές κίνησης σε μικρότερους ή χωμάτινους δρόμους κτλ.

### **3)Περιορισμούς σχετικούς με τους πελάτες**

- Οι πελάτες έχουν συγκεκριμένες απαιτήσεις για προϊόντα που πρέπει να παραδοθούν (π.χ. τροφοδοσία καταστημάτων) ή για υλικά που πρέπει να συλλεχθούν από αυτούς (π.χ. απορρίμματα). Υπάρχει και η περίπτωση να απαιτούν συλλογή και παράδοση ταυτόχρονα (παραλαβή και επιστροφή προϊόντων). Μερικές φορές οι ποσότητες είναι γνωστές επακριβώς και διευκολύνεται ο προγραμματισμός, ενώ αντίθετα αν υπάρχει μερική γνώση των ποσοτήτων, δυσκολεύει η επίλυση του προβλήματος. Σε κάποιες περιπτώσεις, ικανοποιούνται πρώτα από κάθε όχημα οι διανομές και στη συνέχεια, αφού το όχημα έχει ήδη αδειάσει, αναλαμβάνει παραλαβές και μεταφέρει προϊόντα από κάποιους πελάτες στην κεντρική αποθήκη. Στις απλές περιπτώσεις υπάρχει μόνο μια επιστροφή σε κάθε διαδρομή, ενώ στις πιο πολύπλοκες υπάρχουν περισσότερες από μία επιστροφές στο κάθε δρομολόγιο. Ακόμα, υπάρχουν και περιπτώσεις όπου ένα συγκεκριμένο όχημα πρέπει να παραλάβει κάποια προϊόντα από ένα σημείο και να τα παραδώσει σε κάποιο άλλο, διαφορετικό από την κύρια αποθήκη.
- Κάθε πελάτης εξυπηρετείται σε συγκεκριμένα χρονικά διαστήματα (χρονικά παράθυρα). Περιορισμοί, που σχετίζονται με τον χρόνο επίσκεψης, περιλαμβάνονται στα περισσότερα επιχειρησιακά προβλήματα. Σε αυτή την περίπτωση, τα οχήματα θα πρέπει να εξυπηρετήσουν κάποια ή όλα τα σημεία του δικτύου μεταξύ συγκεκριμένων χρονικών ορίων, που αποτελούν το χρονικό παράθυρο (time window). Τα χρονικά παράθυρα είναι χρήσιμα στους πελάτες,



αφού τους βοηθάνε να προγραμματίσουν τις εργασίες τους. Για παράδειγμα, ένα κατάστημα λιανικής πώλησης επιθυμεί να προμηθευτεί τα εμπορεύματα νωρίς το πρωί, πριν αρχίσουν να τα ζητούν οι πελάτες. Από την αντίθετη σκοπιά, αυτές οι πρακτικές είναι δεσμευτικές για την εταιρεία που παραδίδει προϊόντα, περιορίζουν την ευελιξία της και δυσκολεύουν τη λύση του προβλήματος δρομολόγησης στόλου οχημάτων.

- Ο χρόνος εξυπηρέτησης, δηλαδή παραμονής του οχήματος σε κάθε πελάτη, συνήθως είναι σταθερός. Σε πολλές περιπτώσεις όμως, εξαρτάται από τον πελάτη, από την ποσότητα παραγγελίας, από την ευκολία πρόσβασης του οχήματος και άλλους παράγοντες.
- Υπάρχουν προβλήματα όπου δεν μπορούν το σύνολο των οχημάτων να εξυπηρετήσουν το σύνολο των πελατών, καθώς έχουμε περιορισμούς προσέγγισης στα σημεία παράδοσης (π.χ. όγκος οχήματος). Σε πραγματικά προβλήματα δρομολόγησης υπάρχουν λόγοι που επιβάλουν την προτίμηση κάποιων πελατών σε κάποια συγκεκριμένα οχήματα (οδηγούς), λόγω της σχέσης που έχουν αναπτύξει, αλλά και λόγω των ιδιοτεροτήτων των πελατών που γνωρίζει ο οδηγός.
- Υπάρχουν ασυμβατότητες μεταξύ πελατών. Για παράδειγμα, λόγω ανταγωνισμού μεταξύ γειτονικών πελατών, πολλές φορές η επιχείρηση προσπαθεί να τους εξυπηρετεί από διαφορετικά δρομολόγια.
- Οι πελάτες εξυπηρετούνται με σειρά προτεραιότητας. Μπορεί σε κάποιες περιπτώσεις να μην επαρκούν τα προϊόντα ή τα οχήματα και να πρέπει να γίνει επιλογή πελατών που θα ικανοποιηθούν, με βάση κριτήρια που θέτει η εταιρεία.
- Ο πελάτης μπορεί να αποδεχτεί ακόμα και διασπαρμένες παραδόσεις ή παραλαβές, από διαφορετικά οχήματα.

#### **4) Άλλοι παράγοντες**

- Ταξίδια με το όχημα πολλών ωρών, με ολονύχτιες στάσεις.
- Οχήματα χωρισμένα σε διαμερίσματα, με διαφορετικούς τύπους προϊόντων για παράδοση σε κάθε διαμέρισμα.
- Ύπαρξη πολλών αποθηκών και ολοκλήρωση του δρομολογίου σε διαφορετική αποθήκη από την αφετηρία.
- Πολλαπλοί ανεφοδιασμοί την ίδια ημέρα από το ίδιο όχημα.
- Για την μοντελοποίηση της πραγματικότητας με τον καλύτερο δυνατό τρόπο θα πρέπει να ληφθεί υπόψη ότι οι χρόνοι μεταφοράς

εξαρτώνται από εξωγενείς παράγοντες. Επί παραδείγματι, μπορεί τις μεσημεριανές ώρες, ο απαιτούμενος χρόνος για την κάλυψη μίας απόστασης, να είναι πολύ μεγαλύτερος από το χρόνο που απαιτείται για την κάλυψη της ίδιας απόστασης τις βραδινές ώρες.

## 2.3 Στόχοι Δρομολόγησης

Όπως παρουσιάσαμε και νωρίτερα, το VRP είναι ένα πρόβλημα σύνθετου μαθηματικού προγραμματισμού. Αντικειμενική συνάρτηση του προβλήματος είναι το κόστος μεταφοράς, το οποίο θέλουμε να ελαχιστοποιήσουμε, ενώ πρέπει να τηρούνται και οι περιορισμοί. Θα μπορούσαμε να καταλήξουμε λοιπόν σε κάποιους βασικούς στόχους της δρομολόγησης :

- ❖ Η ελαχιστοποίηση του συνολικού στόλου οχημάτων, είτε σε αριθμό είτε σε σύνολο χωρητικότητας, οδηγεί σε οικονομίες κλίμακας, ενώ πρέπει να συνυπολογίζεται η εξοικονόμηση από τους λιγότερους οδηγούς και τις μικρότερες εγκαταστάσεις, τα οποία είναι κατά βάση σταθερά κόστη.
- ❖ Η ελαχιστοποίηση της απόστασης (ή του χρόνου) που συνολικά διανύεται από τα οχήματα, εκτός ότι συντελεί στο στόχο μείωσης του στόλου, αντιπροσωπεύει και τη μείωση του μεταβλητού κόστους για την κίνηση.

Το ιδανικό είναι να μπορέσει η εταιρεία, συνδυάζοντας τη μείωση οχημάτων και τη μείωση της απόστασης, να πετύχει το βέλτιστο μεταφορικό κόστος, λαμβάνοντας υπόψη και όλους τους περιορισμούς. Το κόστος διατήρησης του στόλου οχημάτων, στα περισσότερα προβλήματα, είναι σταθερό κόστος, ενώ το κόστος για την κίνησή τους στις διαδρομές είναι μεταβλητό κόστος. Όπως γίνεται αντιληπτό, η ελαχιστοποίηση και των δύο οδηγεί σε ελαχιστοποίηση του συνολικού κόστους.

Προφανώς, η εταιρεία οφείλει να ικανοποιεί τους πελάτες της στο βαθμό που ορίζει η πολιτική της, δηλαδή μπορεί να υπάρχει ο στόχος ικανοποίησης του συνόλου της ζήτησης ή ενός μεγάλου τμήματος της ζήτησης ή ακόμα και κάποιων σημαντικών πελατών. Το λογισμικό, οι τεχνολογίες και οι αλγόριθμοι που χρησιμοποιεί η επιχείρηση για τη βέλτιστη δρομολόγηση, θα πρέπει να συντονίζουν κάθε λεπτομέρεια της διαδικασίας διανομής ώστε να πετυχαίνονται τα ακόλουθα :

- Τα δεδομένα, που χρησιμοποιούνται, να είναι πραγματικά και να προέρχονται τόσο από τους πελάτες όσο και από την εταιρεία.
- Να προτείνονται τα δρομολόγια χαμηλότερου κόστους συνυπολογίζοντας κάθε παράγοντα, όπως για παράδειγμα την ποιότητα του οδικού δικτύου.
- Να αξιοποιείται στο μέγιστο το προσωπικό και ο μηχανολογικός εξοπλισμός.
- Να βελτιώνεται το επίπεδο εξυπηρέτησης πελατών.
- Να μειώνεται το κόστος διανομής.
- Να ελαττώνεται η επιβάρυνση του περιβάλλοντος (καυσαέρια).
- Λαμβάνονται υπόψη και άλλοι παράγοντες που δεν έχουν σχέση με το οικονομικό κόστος, όπως η ασφάλεια των μεταφερομένων, ή ο κίνδυνος προκλήσεως ατυχήματος κτλ.

Συνεπώς, η αποτελεσματική δρομολόγηση είναι ένα εργαλείο λήψης αποφάσεων, προγραμματισμού λειτουργιών και ανάπτυξης της εταιρικής στρατηγικής, ώστε να αυξηθεί η αποδοτικότητα και τα κέρδη. Η μείωση του λειτουργικού κόστους συνδυάζεται με αύξηση της παραγωγικότητας. Το λειτουργικό κόστος ελαττώνεται με τη μείωση των χιλιομετρικών αποστάσεων, που συνεπάγεται μείωση στα έξοδα συντήρησης, ανταλλακτικών, καυσίμων, υπερωριών κ.α. Επιπλέον, με καλύτερη κατανομή του έργου πετυχαίνεται η χρησιμοποίηση λιγότερων οχημάτων και πρόσθετη εξοικονόμηση μισθών και κεφαλαίων. Ταυτόχρονα η επιχείρηση οδηγείται σε αυξημένη παραγωγικότητα, αφού ελαττώνεται ο χρόνος παραδόσεων και παραλαβών, χρησιμοποιούνται βέλτιστες διαδρομές και ελάχιστοι πόροι (οχήματα, προσωπικό), γίνεται δυναμικός προγραμματισμός δρομολογίων, καλύπτονται έκτακτες ανάγκες και απρόβλεπτες εξελίξεις, ικανοποιούνται οι σημαντικοί πελάτες κτλ.

## 2.4 Το Πρόβλημα του Πλανόδιου Πωλητή

Το πλέον θεμελιώδες πρόβλημα προγραμματισμού διανομής είναι αυτό του πλανόδιου πωλητή (ΠΠΠ) – Traveling Salesman Problem (TSP), στο οποίο ο πωλητής επισκέπτεται πολλαπλές πόλεις (σημεία πώλησης) και επιστρέφει στο σημείο εκκίνησης. Το πρόβλημα αυτό συνίσταται στην εύρεση μίας μόνο διαδρομής που να συνδέει τις  $k$  πόλεις (κόμβους), με τέτοιο τρόπο, ώστε να ελαχιστοποιείται η συνολική διανυθείσα απόσταση ή ο συνολικός χρόνος. Κάθε σημείο πρέπει να δέχεται επίσκεψη μόνο μία φορά. Αν θεωρήσουμε:

- Πόλεις :  $1, 2, \dots, n$  (συμβολίζονται με  $i, j$ )
- $d_{ij}$  την απόσταση από την πόλη  $i$  στην  $j$
- Μεταβλητή απόφασης  $x_{ij}$  όπου,  $x_{ij} = 1$  αν ο πωλητής πηγαίνει από την πόλη  $i$  στην  $j$  αλλιώς  $x_{ij} = 0$

Η αντικειμενική συνάρτηση, εκφράζει τη συνολική απόσταση που διανύει ο πωλητής και είναι:

$$\min \sum_{i,j=1, n, i \neq j} x_{ij} d_{ij}$$

$$x_{ij} \in \{0, 1\}, \text{ για } i \neq j \sum_{j=1, n} x_{ij} = 1 \text{ για } i=1, 2, \dots, n \text{ και } \sum_{i=1, n} x_{ij} = 1 \text{ για } j=1, 2, \dots, n$$

Παρά την ιδιαίτερα απλή διατύπωσή του προβλήματος, είναι πρακτικά αδύνατη η αναλυτική επίλυση του. Αν αντιστοιχιστεί ένα μονοπάτι για κάθε ζευγάρι σημείων, ο συνολικός αριθμός δυνατών λύσεων για  $n$  σημεία είναι  $n!/2$ . Αποτέλεσμα είναι να μην μπορεί να ευρεθεί η βέλτιστη λύση για το πρόβλημα, ακόμα και για μικρές σχετικά τιμές του  $n$ . Συνεπώς, έχουν προταθεί πολλοί ευρετικοί αλγόριθμοι και το TSP έχει αντιμετωπισθεί εκτενώς, με πληθώρα μεθόδων επίλυσης. Ο πιο απλός, μεταξύ των ευρετικών αλγορίθμων, θεωρείται του Πλησιέστερου Γείτονα (Nearest Neighbor). Ξεκινώντας από το σημείο εκκίνησης, επιλέγεται ως επόμενο σημείο επίσκεψης αυτό με την ελάχιστη απόσταση από το τρέχον σημείο. Η διαδικασία αυτή επαναλαμβάνεται μέχρι ο πωλητής να επιστρέψει στο σημείο εκκίνησης, έχοντας επισκεφτεί το σύνολο των σημείων.

## 2.5 Το Πρόβλημα Δρομολόγησης Στόλου Οχημάτων

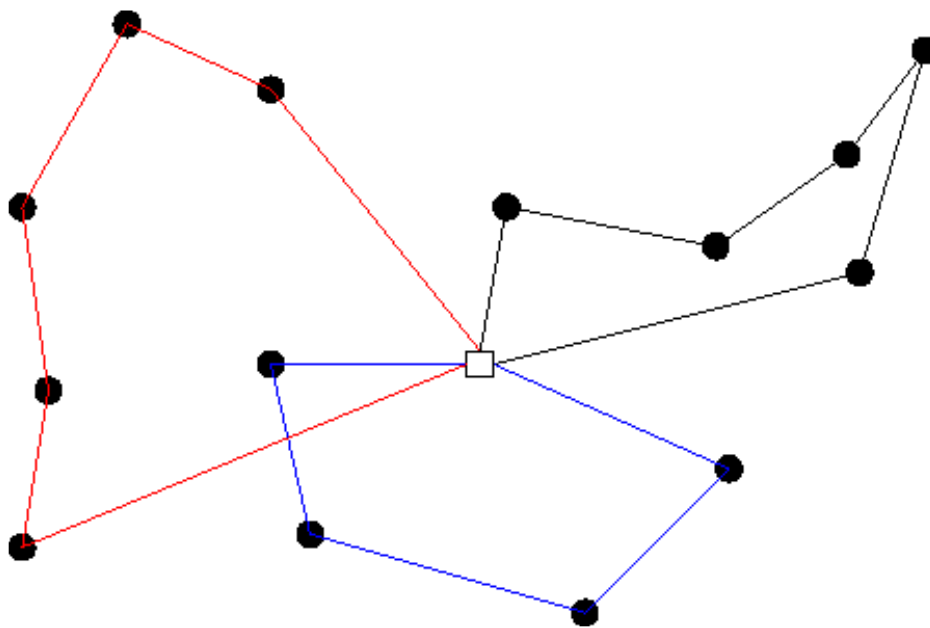
Η γενίκευση του προβλήματος Περιοδεύοντος Πωλητή είναι το πρόβλημα της Δρομολόγησης Στόλου Οχημάτων (Vehicle Routing Problem, VRP). Στο πρόβλημα αυτό, οχήματα γνωστής χωρητικότητας εκκινούν από κοινό σημείο, επισκέπτονται πολλαπλά σημεία πώλησης και επιστρέφουν στο σημείο εκκίνησης. Ο αντικειμενικός σκοπός του προβλήματος είναι εύρεση των στάσεων και της αλληλουχίας επίσκεψης τους από κάθε όχημα, που να ελαχιστοποιεί το συνολικό χρόνο του ταξιδιού ή την απόσταση για το σύνολο του στόλου. Το πρόβλημα ορίζεται σε δίκτυο όπου:

I. Τα τόξα αντιπροσωπεύουν τις διαδρομές από κάθε στάση σε κάθε άλλη στάση και από κάθε στάση στο σημείο εκκίνησης/τερματισμού.

II. Οι κόμβοι αντιπροσωπεύουν τους πελάτες και το σημείο εκκίνησης/τερματισμού.

Κάθε τόξο είναι κατευθυντικό ή μη, ανάλογα με το αν το όχημα μπορεί να το διασχίσει μόνο προς την μία ή και προς τις δύο κατευθύνσεις (οδοί μονής και διπλής κατεύθυνσης). Τα τόξα επίσης χαρακτηρίζονται από το κόστος διέλευσης, το οποίο συνήθως αντιπροσωπεύεται από το μέτρο του τόξου ως απόσταση.

Η απλή διατύπωση του προβλήματος είναι : έστω  $n$  οχήματα διαθέσιμα, χωρητικότητας  $V$  και έστω  $d_i$  ( $i=1..m$ ) τα φορτία, που απαιτείται να διανεμηθούν σε  $m$  κόμβους πελάτες  $P_j$  ( $j=1..m$ ) από ένα αρχικό κόμβο αποθήκη  $P_0$ . Δεδομένων των αποστάσεων (ή κόστους)  $c_{yz}$  μεταξύ των κόμβων, ζητείται να υπολογιστούν  $k$  κυκλικές διαδρομές, όπου  $k \leq m$ , που ξεκινούν και καταλήγουν στο  $P_0$  και εξυπηρετούν τους  $m$  κόμβους, ελαχιστοποιώντας τη συνολική διανυόμενη απόσταση από τα οχήματα. Άρα, θα πρέπει να οριστούν έτσι οι κυκλικές διαδρομές ώστε να εξυπηρετείται η ζήτηση των πελατών στον ελάχιστο χρόνο (κόστος), ενώ ταυτόχρονα να μην υπερβαίνεται η χωρητικότητα των  $n$  οχημάτων.



Εικόνα 5 Κυκλικές διαδρομές για εξυπηρέτηση πελατών (κόμβων).

## 2.6 Κατηγορίες προβλημάτων δρομολόγησης οχημάτων(VRP)

### **Προβλήματα Δρομολόγησης Οχημάτων περιορισμένης χωρητικότητας (Capacitated Vehicle Routing Problem – CVRP).**

Η συνηθέστερη μορφή των προβλημάτων δρομολόγησης είναι τα «Προβλήματα Δρομολόγησης Οχημάτων με Συγκεκριμένο Όριο Ικανότητας» (CVRP). Σε αυτές τις περιπτώσεις υπάρχει ένας συγκεκριμένος αριθμός οχημάτων, καθορισμένης χωρητικότητας, που εξυπηρετεί τις ήδη γνωστές απαιτήσεις των πελατών για προϊόντα από μία αποθήκη. Στόχος είναι να γίνουν οι παραδόσεις στο ελάχιστο δυνατό κόστος. Στην ουσία, μιλάμε για ένα κλασσικό πρόβλημα VRP με τη διαφορά ότι κάθε όχημα έχει ανώτατο επίπεδο χωρητικότητας. Διατυπώνοντας καλύτερα ένα τέτοιο πρόβλημα έχουμε:

Οι  $n$  πελάτες εξυπηρετούνται από μια αποθήκη. Ο κάθε πελάτης απορροφά ποσότητα  $q_i$  προϊόντων ( $i = 1, 2, \dots, n$ ). Στη διάθεσή μας υπάρχει ένας αριθμός οχημάτων χωρητικότητας  $Q$ . Από μαθηματικής άποψης, σε σχέση με ένα VRP πρόβλημα, προστίθεται ο περιορισμός ότι η συνολική απαίτηση των πελατών, που εξυπηρετούνται από μία διαδρομή  $R_i$ , δεν πρέπει να υπερβαίνει τη χωρητικότητα των οχημάτων  $Q$ :  $\sum_{i \in R_i} q_i \leq Q$ .

Τα οχήματα, αφού έχουν περιορισμένη χωρητικότητα, θα πρέπει να επιστρέφουν περιοδικά στην αποθήκη για να ξαναφορτώσουν. Ο πελάτης δεν εξυπηρετείται από διαφορετικά οχήματα, άρα σε μια εφικτή λύση ο

πελάτης δέχεται επίσκεψη μόνο μια φορά, ενώ η μέγιστη απαίτηση σε προϊόντα μιας διαδρομής δεν πρέπει να υπερβαίνει τη χωρητικότητα  $Q$  του οχήματος που την εκτελεί. Κάθε διαδρομή αρχίζει και ολοκληρώνεται στην αποθήκη, ενώ στόχος είναι η εύρεση του συνδυασμού των διαδρομών που ελαχιστοποιεί το συνολικό κόστος. Αυτό πετυχαίνεται μέσα από την ελαχιστοποίηση τόσο του συνολικού στόλου οχημάτων, όσο και του συνολικού χρόνου ταξιδιού.

### **Προβλήματα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (Vehicle Routing Problem with Time Windows – VRPTW).**

Το πρόβλημα αυτό, εκτός από τα προαναφερθέντα χαρακτηριστικά ενός CVRP προβλήματος, περιλαμβάνει, για την αποθήκη αλλά και για κάθε πελάτη  $c_i$  ( $i = 0, \dots, n$ ), ένα χρονικό παράθυρο  $[b_i, e_i]$  κατά τη διάρκεια του οποίου ο κάθε πελάτης θα πρέπει να εξυπηρετηθεί. Ο νωρίτερος χρόνος εκκίνησης από την αποθήκη είναι  $b_0$  και ο αργότερος χρόνος επιστροφής στην αποθήκη  $e_0$ . Κάθε κόμβος πελάτης εξυπηρετείται από το όχημα μέσα στο χρονικό παράθυρο. Σε περίπτωση που το όχημα φτάσει στον κόμβο πριν από το νωρίτερο χρόνο  $b_i$ , τότε είναι υποχρεωμένο να περιμένει μέχρι τη χρονική στιγμή που ορίζει το παράθυρο. Ο χρόνος αναχώρησης από πελάτη  $c_i$  πρέπει να είναι μικρότερος ή ίσος από το χρόνο  $e_i$ . Ο αριθμός  $u$  των διαθέσιμων οχημάτων πολλές φορές πρέπει να υπολογιστεί. Το κόστος μιας διαδρομής  $i$  υπολογίζεται από τον τύπο  $CVRPTW(R_i) = \sum c_{i,i+1} + \sum d_{i,i+1} + \sum w u_i$  όπου  $i=(0,m)$  και ο χρόνος αναμονής του πελάτη  $u_i$  είναι  $w u_i$ .

Στόχος ενός τέτοιου προβλήματος είναι η ελαχιστοποίηση του στόλου οχημάτων, η ελαχιστοποίηση του χρόνου ταξιδιού και των χρόνων αναμονής για την εξυπηρέτηση των πελατών.

Ως προς την εφικτότητα, το VRPTW σε σχέση με το VRP χαρακτηρίζεται από τους ακόλουθους επιπλέον περιορισμούς :

- Μια λύση είναι μη εφικτή αν ο πελάτης εξυπηρετείται μετά το άνω όριο του χρονικού παραθύρου.
- Άφιξη οχήματος σε πελάτη πριν από το κάτω όριο του χρονικού παραθύρου, οδηγεί σε επιπρόσθετο χρόνο αναμονής.
- Κάθε διαδρομή αρχίζει και ολοκληρώνεται εντός του χρονικού παραθύρου της αποθήκης.
- Όταν υπάρχουν ευέλικτα χρονικά παράθυρα, η εξυπηρέτηση ενός πελάτη με καθυστέρηση δεν επηρεάζει την εφικτότητα της λύσης, απλά καθυστερεί επόμενες παραδόσεις.

### **Προβλήματα Δρομολόγησης Οχημάτων με παραδόσεις και διανομές (Vehicle Routing Problem with Pickup and Delivery – VRPPD).**

Πρόκειται για ένα VRP πρόβλημα, στο οποίο προβλέπεται ότι οι πελάτες επιστρέφουν κάποια προϊόντα. Συνεπώς, πρέπει να προβλεφθεί χώρος στο όχημα για τα επιστρεφόμενα προϊόντα, καθιστώντας το πρόβλημα αρκετά περίπλοκο ως προς την ορθή χρησιμοποίηση της χωρητικότητας των οχημάτων. Στα VRPPD όλες οι παραγγελίες ξεκινούν από την αποθήκη και όλα τα επιστρεφόμενα επιστρέφουν στην αποθήκη, δεν υπάρχει ανταλλαγή προϊόντων μεταξύ πελατών και οι πελάτες δέχονται μόνο μια επίσκεψη. Σε μια απλοποιημένη μορφή του προβλήματος, το όχημα παραδίδει όλες τις παραγγελίες πριν δεχτεί επιστροφές.

Στόχος είναι η ελαχιστοποίηση του αριθμού των χρησιμοποιηθέντων οχημάτων και του χρόνου των διαδρομών σε συνδυασμό με την απόλυτη κάλυψη τόσο των αναγκών παραδόσεων όσο και επιστροφών.

Μια λύση είναι εφικτή εφόσον η συνολικά μεταφερόμενη ποσότητα από κάθε όχημα, συμπεριλαμβανομένων και των επιστροφών, δεν υπερβαίνει τη χωρητικότητα του οχήματος.

### **Προβλήματα Δρομολόγησης Οχημάτων με Επιστροφές (Vehicle Routing Problem with Backhauls – VRPB).**

Σε αυτά τα προβλήματα, όπως και στα VRPPD, οι πελάτες έχουν τη δυνατότητα, εκτός από το να παραλαμβάνουν, να επιστρέφουν και κάποια προϊόντα. Η διαφορά των δύο προβλημάτων είναι ότι στο VRPPD απαιτείται να γίνει πρόβλεψη ώστε οι επιστροφές να χωρούν στα οχήματα, αντίθετα στο VRPB όλες οι παραδόσεις γίνονται πριν από τις επιστροφές. Με αυτή τη μέθοδο αποφεύγεται η αντισυμβατική αναδιοργάνωση των φορτίων στα σημεία παράδοσης. Οι ποσότητες των προϊόντων που παραδίδονται και επιστρέφονται είναι εκ των προτέρων γνωστές. Συνεπώς, στη διατύπωση του προβλήματος υπάρχουν περιορισμοί από τον αριθμό των παραδόσεων και επιστροφών και τον όγκο του φορτίου. Στόχος είναι ο σχεδιασμός της βέλτιστης διαδρομής που θα ελαχιστοποιήσει τη συνολικά διανυόμενη απόσταση, ενώ σε μια εφικτή λύση υπάρχει ένα σύνολο διαδρομών, όπου όλες οι παραδόσεις για κάθε διαδρομή γίνονται πριν αρχίσουν επιστροφές.



### **Προβλήματα Δρομολόγησης Οχημάτων με πολλαπλές Αποθήκες (Vehicle Routing Problem with Multiple Depot – MDVRP).**

Στις περιπτώσεις αυτές, οι πελάτες εξυπηρετούνται από περισσότερες από μία αποθήκες. Όταν όλοι οι πελάτες είναι συγκεντρωμένοι γύρω από τις αποθήκες, τότε το πρόβλημα μπορεί να εξεταστεί ως συνδυασμός ανεξάρτητων VRP προβλημάτων. Όμως στις περισσότερες περιπτώσεις οι πελάτες δεν είναι ομοιόμορφα κατανομημένοι γύρω από τις αποθήκες και τότε λύνεται ως MDVRP. Γίνεται ανάθεση των πελατών σε διάφορες αποθήκες, που έχουν δικούς τους στόλους οχημάτων, και κάθε όχημα, που έχει αφετηρία μια αποθήκη, επιστρέφει στο τέλος σε αυτή. Στόχος είναι η ελαχιστοποίηση του αριθμού των χρησιμοποιούμενων οχημάτων και της διανυόμενης απόστασης, ενώ οι διάφορες αποθήκες πρέπει να ικανοποιήσουν τη συνολική απαίτηση των πελατών. Λέμε μια λύση εφικτή, όταν ικανοποιούνται οι περιορισμοί ενός VRP προβλήματος και τα οχήματα αρχίζουν και επιστρέφουν στην ίδια αποθήκη.

### **Προβλήματα Δρομολόγησης Οχημάτων κατά Περίοδο (Periodic Vehicle Routing Problem– PVRP).**

Αντίθετα με τα συνηθισμένα προβλήματα στα οποία ρυθμίζονται οι παραδόσεις για μία ημέρα, το PVRP επεκτείνει το διάστημα προγραμματισμού των διανομών σε  $M$  ημέρες. Είναι ένα πολύ-επίπεδο συνδυαστικό πρόβλημα βελτιστοποίησης όπου σε πρώτο επίπεδο πρέπει να καθοριστεί η περίοδος με την οποία γίνονται οι διανομές σε κάθε πελάτη (π.χ. κάθε δύο ημέρες), σε δεύτερο επίπεδο επιλέγονται οι εναλλακτικές λύσεις για κάθε πελάτη και σε τρίτο επίπεδο λύνεται ένα VRP πρόβλημα για κάθε ημέρα της περιόδου.

Στόχος είναι τόσο η ελαχιστοποίηση του συνολικού αριθμού οχημάτων όσο και του συνολικού χρόνου ταξιδιού για την εξυπηρέτηση των πελατών. Μία λύση θεωρείται εφικτή εφόσον καλύπτονται όλοι οι περιορισμοί του προβλήματος VRP, με τη διαφορά ότι ένα όχημα μπορεί να μην επιστρέψει στην αποθήκη την ίδια ημέρα με αυτή που αποχώρησε από την αποθήκη. Επιπλέον, στις  $M$  ημέρες της περιόδου διανομής, κάθε πελάτης δέχεται τουλάχιστον μία επίσκεψη.

### **Στοχαστικά – Πιθανολογικά Προβλήματα Δρομολόγησης Οχημάτων (Stochastic Vehicle Routing Problem– SVRP).**

Στα SVRP προβλήματα, ένα ή περισσότερα στοιχεία του προβλήματος (π.χ. πελάτες, απαιτήσεις σε εμπορεύματα) δεν θεωρούνται σταθερά. Ενδεικτικές περιπτώσεις SVRP είναι :

- Στοχαστικοί πελάτες : Η πιθανότητα κάθε πελάτη  $i$  να πραγματοποιεί παραγγελία είναι  $p_i$ , ενώ δεν πραγματοποιείται παραγγελία με πιθανότητα  $1 - p_i$ .
- Στοχαστικές απαιτήσεις πελατών : Η ζήτηση  $d_i$  κάθε πελάτη δεν είναι σταθερή.
- Στοχαστικό χρόνος ταξιδιού : Ο χρόνος, που απαιτείται για την προσέγγιση ενός πελάτη, αλλά και ο χρόνος για την εξυπηρέτησή του, δεν είναι σταθερός.

Στα SVRP προβλήματα υπάρχουν δύο στάδια επίλυσης. Στην αρχή, δίνεται μια πρώτη λύση πριν γίνουν γνωστές οι τυχαίες μεταβλητές, ενώ στη συνέχεια, γνωρίζοντας τις τιμές των τυχαίων μεταβλητών, διορθώνεται η πρώτη λύση. Στόχος είναι τόσο η ελαχιστοποίηση του συνολικού αριθμού οχημάτων όσο και του συνολικού χρόνου ταξιδιού για την εξυπηρέτηση των πελατών. Επειδή οι τιμές κάποιων μεταβλητών δεν είναι σταθερές, αλλά μεταβάλλονται συνέχεια, δεν μπορούν να ικανοποιούνται όλοι οι περιορισμοί για κάθε πιθανή τιμή. Ο υπεύθυνος δρομολόγησης έχει λοιπόν δύο επιλογές :

- Να ικανοποιήσει κάποιους από τους περιορισμούς με μια δεδομένη πιθανότητα.
- Να ενσωματώσει ένα πρότυπο διορθωτικών ενεργειών, που να ενεργοποιείται όταν δεν ικανοποιείται κάποιος περιορισμός

### **Προβλήματα Δρομολόγησης Οχημάτων με εξυπηρέτηση πελάτη από άνω του ενός οχήματος (Split Delivery Vehicle Routing Problem – SDVRP).**

Πρόκειται για μια παραλλαγή του VRP προβλήματος στην οποία κάθε πελάτης δύναται να εξυπηρετηθεί από περισσότερα του ενός οχήματα, εφόσον έτσι πετυχαίνονται μειωμένα κόστη διανομής. Σε περιπτώσεις που ο όγκος παραγγελιών του πελάτη είναι μεγαλύτερος από τη χωρητικότητα του οχήματος, το SDVRP δίνει πραγματικές λύσεις. Στο SDVRP γίνεται κατάτμηση των παραδόσεων, με διάσπαση της παραγγελίας του πελάτη, για να επιτευχθεί στη συνέχεια ελαχιστοποίηση του κόστους παραδόσεων. Στόχος είναι η ελαχιστοποίηση του συνολικού αριθμού οχημάτων που χρησιμοποιούνται και η ελαχιστοποίηση του συνολικού κόστους διαδρομών για την εξυπηρέτηση των πελατών. Η εφικτή λύση υπάρχει όταν ικανοποιούνται οι περιορισμοί του VRP, με τη διαφοροποίηση της εξυπηρέτησης του πελάτη από περισσότερα του ενός οχήματα.

## **Άλλα Προβλήματα Δρομολόγησης Οχημάτων**

Πιο συνοπτικά παρουσιάζουμε και κάποια άλλες παραλλαγές του VRP.

- Χρόνο εξαρτώμενη Δρομολόγηση Στόλου Οχημάτων (Time Dependent Vehicle Routing Problem – TDVRP) : Λαμβάνεται υπόψη η κίνηση του δικτύου κάθε στιγμή. Άρα, ο χρόνος κίνησης μεταξύ δύο πελατών μεταβάλλεται.
- Δρομολόγηση Ετερογενούς Στόλου Οχημάτων (Heterogeneous Fleet Vehicle Routing Problem – HFVRP) : Υπάρχει στόλος οχημάτων με διαφορετικά χαρακτηριστικά. Η διαφορά μπορεί να βρίσκεται στη χωρητικότητα των οχημάτων, στο κόστος λειτουργίας, στην τεχνολογία και σε ποικίλους άλλους τομείς.
- Δρομολόγηση Στόλου Οχημάτων με Εξισορρόπηση Διαδρομών (Vehicle Routing Problem with Route Balancing – VRPRB) : Σε αυτά τα προβλήματα πρέπει να μειωθεί η διαφορά μεταξύ της μεγαλύτερης και της μικρότερης σε μήκος διαδρομής. Αυτός ο περιορισμός είναι σημαντικός για την ίση κατανομή φόρτου εργασίας μεταξύ των οδηγών.
- Δυναμική Δρομολόγηση Στόλου Οχημάτων (Dynamic Vehicle Routing Problem – DVRP) : Ο πελάτης δύναται να ακυρώνει παραγγελίες ή να ενσωματώνει νέες παραγγελίες πέρα από τις ήδη γνωστές.

## Κεφάλαιο 3:Αλγόριθμοι επίλυσης προβλημάτων

### διανομής

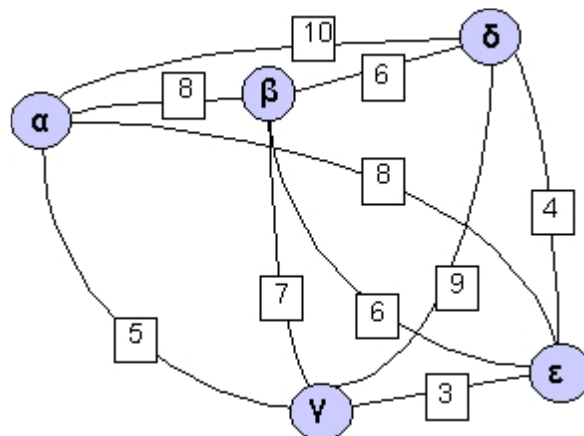
Λόγω του γεγονότος ότι το VRP και όλες οι παραλλαγές του είναι προβλήματα της κλάσης NP-Hard, η επίλυσή τους είναι ιδιαίτερα δύσκολη και σε ορισμένες περιπτώσεις η ακριβής επίλυση μέσω ενός εξονυχιστικού ελέγχου όλων των πιθανών διαδρομών είναι ανέφικτη. Για τον λόγο αυτό έχουν αναπτυχθεί ορισμένοι αλγόριθμοι που δεν βρίσκουν την βέλτιστη λύση αλλά λύση που προσεγγίζει την βέλτιστη παράγοντας μια ικανοποιητική διαδρομή. Οι βασικές κατηγορίες των αλγορίθμων αυτών βάσει του αν είναι ακριβείς ή προσεγγιστικοί αλγόριθμοι και βάσει του χρόνου επίλυσης είναι οι ακόλουθοι.

### 3.1 Ακριβείς αλγόριθμοι

Οι ακριβείς αλγόριθμοι εύρεσης της βέλτιστης λύσης ελέγχουν συστηματικά όλες τις πιθανές διαδρομές και επιλέγουν την καλύτερη. Παρότι είναι οι μόνοι αλγόριθμοι που βρίσκουν την βέλτιστη λύση όσο μεγαλώνει το μέγεθος του προβλήματος τόσο αυξάνει και ο χρόνος επίλυσής του φτάνοντας έτσι σε μεγάλα προβλήματα μη αποδεκτό χρόνο επίλυσης του προβλήματος. Μέχρι στιγμής οι καλύτεροι ακριβείς αλγόριθμοι περιορίζονται στο να λύνουν προβλήματα μέχρι 100 πελάτες Σε αυτή την κατηγορία αλγορίθμων ανήκουν οι branch and bound και branch and cut αλγόριθμοι.

#### 3.1.1 Αλγόριθμος Branch and Bound'

Ο αλγόριθμος επέκτασης και οριοθέτησης (Branch and Bound - B&B) εφαρμόζεται σε προβλήματα όπου αναζητείται η βέλτιστη λύση, δηλαδή εκείνη με το ελάχιστο κόστος.



Εικόνα 6: Αναπαράσταση αρχικού χώρου αναζήτησης BranchandBound.

Η λειτουργία του B&B βασίζεται στο κλάδεμα καταστάσεων (pruning) και κατά συνέπεια στην ελάττωση του χώρου αναζήτησης.

**Ουσιαστικά τα βήματα που ακολουθούνται είναι τα εξής:**

1. Βάλε την αρχική κατάσταση στο μέτωπο της αναζήτησης.
2. Αρχική τιμή της καλύτερης λύσης είναι το  $+\infty$  (όριο).
3. Αν το μέτωπο της αναζήτησης είναι κενό, τότε σταμάτησε. Η καλύτερη μέχρι τώρα λύση είναι και η βέλτιστη.
4. Βγάλε την πρώτη σε σειρά κατάσταση από το μέτωπο της αναζήτησης.
5. Αν η κατάσταση ανήκει στο κλειστό σύνολο, τότε πήγαινε στο 3.
6. Αν η κατάσταση είναι τελική, τότε ανανέωσε τη λύση ως την καλύτερη μέχρι τώρα και ανανέωσε την τιμή του ορίου με την τιμή που αντιστοιχεί στην τελική κατάσταση. Πήγαινε στο 3.
7. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά και την τιμή που αντιστοιχεί σε αυτές.
8. Βάλε τις καταστάσεις-παιδιά, των οποίων η τιμή δεν υπερβαίνει το όριο, μπροστά στο μέτωπο της αναζήτησης.
9. Βάλε την κατάσταση-γονέα στο κλειστό σύνολο.
10. Πήγαινε στο 3.

**Ψευδοκώδικας**

*begin*

*Closed* ←  $\emptyset$ ;

*Frontier* ← *<InitialState>*;

*BestCost* ←  $\alpha$ ;

*while Frontier* ≠  $\emptyset$  *do*

*CurrentState* ← *First(Frontier)*;

*CurrentCost* ← *Cost(Current\_State)*;

*Frontier* ← *delete(CurrentState, Frontier)*;

*If CurrentState* ∉ *Closed* *then*

*begin*

*if CurrentState* ∈ *FinalStates* and *CurrentCost* < *BestCost* *then*

*BestCost* ← *CurrentCost*;

*else*

*if CurrentCost* < *BestCost* *then*

*begin*

*ChildrenStates* ← *Expand*(*CurrentState*);

*Frontier* ← *ChildrenStates* ∪ *Frontier*;

*Closed* ← *Closed* ∪ {*CurrentState*};

*end*;

*end*;

*endwhile*;

*end*.

Ο Β&Β μπορεί να συνδυαστεί με δυναμικό προγραμματισμό (dynamic programming), όπου το κλάδεμα δε γίνεται μόνο σε σύγκριση με το τρέχον όριο, δηλαδή τη βέλτιστη λύση μέχρι εκείνη τη στιγμή, αλλά γίνεται και για κάθε κατάσταση που είναι περιττή.

### 3.2 Ευρετικοί αλγόριθμοι (heuristics)

Οι ευρετικοί αλγόριθμοι μειώνουν τον χρόνο επίλυσης του προβλήματος μην παρέχοντας όμως την βέλτιστη λύση αλλά μία καλή λύση που προσεγγίζει την βέλτιστη. Για να είναι ένας ευρετικός αλγόριθμος καλός πρέπει η απόκλιση της λύσης που δίνει να μην είναι μεγάλη από την βέλτιστη λύση, η εύρεση της λύσης να επιτυγχάνεται εύκολα και η λογική στην οποία στηρίζονται οι κανόνες που οδηγούν στην λύση να είναι σωστή. Σε αυτή την κατηγορία αλγορίθμων ανήκουν και οι αλγόριθμοι τοπικής αναζήτησης (local search algorithms).

### 3.2.1 Ο αλγόριθμος του πλησιέστερου γείτονα

Η μέθοδος του Πλησιέστερου Γείτονα αποτελεί μία από τις πρώτες και πλέον απλές μεθόδους για την επίλυση του προβλήματος του Πλανόδιου Πωλητή, η οποία προτάθηκε από τον Δρ. Flood το 1956. Βέβαια, με την εμφάνιση σύνθετων μεθευρετικών πλαισίων και το διαχωρισμό των τοπικών τελεστών αναζήτησης σε κατασκευαστικούς και τελεστές βελτίωσης, η μέθοδος Nearest Neighbor(NN)έχει τοποθετηθεί στο σύνολο των κατασκευαστικών τελεστών. Ιδιαίτερο ενδιαφέρον παρουσιάζει η παρατήρηση ότι κατά μέση απόδοση ο NN παράγει μία διαδρομή 25% μεγαλύτερη από την πλέον σύντομη για ένα καθορισμένο πλήθος πόλεων, τυχαία τοποθετημένων σε ένα επίπεδο.

Για την κατασκευή ενός εφικτής λύσης, η μέθοδος του NN ξεκινά τυχαία από έναν κόμβο του γραφήματος και το συνδέει με τον πλησιέστερο ως αυτόν κόμβο, με τη βασική προϋπόθεση ότι ο τελευταίος δεν έχει ήδη επισκεφθεί. Στη συνέχεια, ο κόμβος που μόλις συνδέθηκε στη σχηματιζόμενη διαδρομή συνδέεται με τον πλησιέστερο ως ενός αυτόν μη συνδεδεμένο κόμβο και η συγκεκριμένη διαδικασία συνεχίζεται μέχρι να ολοκληρωθεί η διαδρομή, δηλαδή να έχουν επισκεφθεί όλοι οι κόμβοι. Με σκοπό την καλύτερη οπτική απόδοση της ανωτέρω περιγραφής, ακολουθεί η λειτουργία της μεθόδου σε βήματα:

- **Βήμα 1:**  
Επιλογή τυχαίου κόμβου ως αφετηρία.
- **Βήμα 2:**  
Μετάβαση στον πλησιέστερο μη συνδεδεμένο κόμβο.
- **Βήμα 3:**  
Αν υπάρχουν κόμβοι που δεν έχουν επισκεφθεί, τότε επαναλαμβάνεται το Βήμα 2.
- **Βήμα 4:**  
Επιστροφή στον αρχικό κόμβο.

### 3.2.2 Αλγόριθμοι Relocate

Η μέθοδος της επανατοποθέτησης αποτελεί μία ακόμα εύκολη ως προς τη λειτουργία και απλή ως προς την κατανόηση δομή γειτνίασης. Η κύρια λειτουργία της αφορά την επιλογή ενός κόμβου, την αφαίρεσή του από την έχουσα κυκλική διαδρομή και την επανατοποθέτησή του στη θέση ενός εκ των υπολοίπων κόμβων της διαδρομής. Βέβαια για την επιτυχή επανατοποθέτηση του κόμβου θα πρέπει να πραγματοποιηθεί μία οπισθοδρομική μετατόπιση ορισμένων στοιχείων κατά μία θέση. Ακριβέστερα, θεωρείται ότι αφαιρείται ο κόμβος  $i$  και ζητείται να επανατοποθετηθεί στη θέση του κόμβου  $j$ , τότε οι κόμβοι από τον  $i+1$  έως και τον κόμβο  $j$ , θα μετατοπισθούν κατά μία θέση οπισθοδρομικά. Πιο συχνοί σε χρήση είναι οι 1-0 και 2-0 relocate.

Για παράδειγμα αν είχαμε την ακόλουθη λύση:

**1 7 4 9 5 8 3 10 2 6 1**

θα μπορούσαμε να πάρουμε ένα κόμβο, για παράδειγμα τον κόμβο 5, και να τον τοποθετούσαμε μεταξύ του 4 και του 9 δηλαδή να τον μεταφέρουμε μία θέση δίπλα στο διάνυσμα και θα είχαμε την ακόλουθη λύση:

**1 7 4 5 9 8 3 10 2 6 1**

Για να δούμε αν αυτή η λύση είναι εφικτή θα πρέπει να εξετάσουμε τη διαφορά του κόστους των τόξων που εισέρχονται με τα τόξα που εξέρχονται. Στην ουσία έχουμε 3 τόξα που εξέρχονται, τα οποία είναι τα τόξα που συνδέονται με τον κόμβο που θέλουμε να μεταφέρουμε, 5-9 και 5-8, και το τόξο που ενώνει τους δύο κόμβους ανάμεσα στους οποίους θα εισέλθει ο κόμβος 5, άρα το τόξο 4-9. Επίσης έχουμε και τρία τόξα που εισέρχονται στη λύση τα τόξα 4-5 και 5-9 και το τόξο 8-9 που ενώνει το κενό που άφησε ο κόμβος 5. Άρα έχουμε  $c_{95} + c_{58} + c_{49} - c_{45} - c_{59} - c_{98}$ . Αν και πάλι έχουμε θετικό πρόσημο σημαίνει ότι έχουμε βελτίωση και η λύση είναι αποδεκτή.

Ο αλγόριθμος 2-0 relocate λειτουργεί ακριβώς το ίδιο με τον αλγόριθμο 1-0 relocate μόνο που αντί να μετακινηθεί ένας κόμβος σε άλλη θέση μετακινούνται δύο. Ο 2-0 Relocate και ο 1-1 Exchange Χρησιμοποιήθηκαν στον κώδικα της διατριβής που θα περιγράψουμε σε επόμενο κεφάλαιο.



### 3.2.3 Αλγόριθμος 2-opt

Μια κίνηση 2-opt διαγράφει δύο άκρα, χωρίζοντας τη διαδρομή σε δύο μέρη, και στη συνέχεια επανασυνδέει τα μονοπάτια με άλλον δυνατό τρόπο, ώστε να έχουμε μείωση του κόστους. Αυτό ισοδυναμεί με αντιστροφή της σειράς των πελατών μεταξύ των δύο άκρων.

Η διαδικασία της μεθόδου 2-opt είναι η ακόλουθη:

#### **Βήμα 1:**

Έστω  $T$  η τρέχουσα διαδρομή.

#### **Βήμα 2:**

Για κάθε κόμβο  $i=1,...,n$ : Εξετάζουμε όλες τις πιθανές 2-opt κινήσεις που μπορεί να γίνουν από τον  $i$  και την επόμενη της μέσα στην διαδρομή. Αν με αυτό τον τρόπο μπορούμε να μειώσουμε το κόστος της διαδρομής, τότε επιλέγουμε την καλύτερη 2-opt κίνηση και εφαρμόζουμε τις αλλαγές στην διαδρομή  $T$ .

#### **Βήμα 3:**

Αν δεν μπορούμε να βρούμε επιπλέον βελτίωση, σταματάμε.

Στην χειρότερη περίπτωση, μπορεί μονάχα να εγгуηθεί ότι μια κίνηση βελτίωσης μειώνει το μήκος της διαδρομής τουλάχιστον μια μονάδα.

Δουλεύουμε και εδώ όπως και στους προηγούμενους αλγορίθμους χρησιμοποιώντας το `temproute`, το `temp1` και το `temp2`.

Για παράδειγμα, σε ένα διάνυσμα

6-5-8-2-9-3-1

αν το  $\text{randi}(7)=3$  και το άλλο  $\text{randi}(7)=6$ , τότε το κελί 3, δηλαδή ο πελάτης 8, θα τοποθετηθεί στη θέση του κόμβου 6, δηλαδή θα γίνει αντιμετάθεση με τον πελάτη 3. Στη συνέχεια γίνεται αντιστροφή όλων των ενδιάμεσων κελιών/πελατών. Οπότε το συγκεκριμένο διάνυσμα θα γίνει:

6-5-3-9-2-8-1

Κατά την ολοκλήρωση του αλγορίθμου τοπικής αναζήτησης 2-opt γίνεται και πάλι έλεγχος του περιορισμού του χρόνου και σύγκριση του κόστους της διαδρομής.

Η επαναληπτική διαδικασία συνεχίζεται για όσες επαναλήψεις προκαθορίσουμε εμείς. Όσο περισσότερες είναι οι επαναλήψεις τόσο αυξάνονται οι πιθανότητες να βρούμε καλύτερο αποτέλεσμα. Στο τέλος κρατάμε τα δρομολόγια που παρουσιάζουν το χαμηλότερο κόστος,

ανεξάρτητα από τη μέθοδο της τοπικής αναζήτησης από την οποία προέκυψε.

#### 3.2.4 Αλγόριθμος 1-1 Exchange

Ένας πολύ γνωστός αλγόριθμος τοπικής αναζήτησης είναι ο αλγόριθμος 1-1 ανταλλαγή (1-1 exchange). Στο συγκεκριμένο αλγόριθμο 2 κόμβοι αλλάζουν θέση μεταξύ τους. Έτσι για το αρχικό τυχαίο παράδειγμα με λύση:

**1 7 4 9 5 8 3 10 2 6 1**

αν αντιμετωπιστούν οι κόμβοι 2 και 8 τότε έχουμε την ακόλουθη λύση:

**1 7 4 9 5 2 3 10 8 6 1**

Σε αυτή την λύση διαγράφονται τα τόξα που συνδέονται με τους κόμβους 2 και 8, άρα τα τόξα 10-2, 2-6, 5-8 και 8-3 και προστίθενται τα τόξα 10-8, 8-6, 5-2 και 2-3. Η διαφορά στο κόστος θα ήταν  $c_{10,2} + c_{2,6} + c_{5,8} + c_{8,3} - c_{10,8} - c_{8,6} - c_{5,2} - c_{2,3}$ . Από τη στιγμή που έχουμε θετικό πρόσθετο σημαίνει ότι έχουμε βελτίωση και η λύση είναι αποδεκτή.

### 3.3 Μεθευρετικοί αλγόριθμοι (Metaheuristics)

Οι μεθευρετικοί αλγόριθμοι όπως και οι ευρετικοί δίνουν λύσεις που είναι κοντά στην βέλτιστη λύση. Οι μεθευρετικοί όμως είναι αλγόριθμοι υψηλού επιπέδου, αποδοτικότεροι από τους ευρετικούς (Yang, 2010c) και μπορούν να χρησιμοποιηθούν για την επίλυση σχεδόν όλων των προβλημάτων βελτιστοποίησης (Talbi, 2009). Τα βασικά χαρακτηριστικά τους είναι η εκμετάλλευση (exploitation) και η εξερεύνηση (exploration). Η εκμετάλλευση αφορά την αναζήτηση σε μία μικρή περιοχή στην οποία έχει ήδη εντοπιστεί μία καλή λύση. Η εξερεύνηση σχετίζεται με την αναζήτηση σε όλο τον χώρο λύσεων με την βοήθεια της τυχειότητας η οποία δεν επιτρέπει στον αλγόριθμο να εγκλωβίζεται σε τοπικά βέλτιστα. Οι μεθευρετικοί αλγόριθμοι μοντελοποιούν κάποιο φαινόμενο που υπάρχει στην φύση και είναι προσαρμοστικοί.

Τέτοιου είδους αλγόριθμοι είναι η προσομοιωμένη ανόπτηση (simulated annealing) η περιορισμένη αναζήτηση (tabu search), οι γενετικοί και εξελικτικοί αλγόριθμοι (genetic and evolutionary algorithms), αλγόριθμοι βελτιστοποίησης αποικίας μυρμηγκιών (ant colony optimization) και άλλοι. Ακολουθούν οι προαναφερθέντες.

#### 3.3.1 Προσομοιωμένη Ανόπτηση (Simulated Annealing)

Οι μέθοδοι προσομοιωμένης ανόπτησης (simulated annealing) βασίζονται σε μια αναλογία με την μεταλλουργική διεργασία της ανόπτησης, κατά την οποία ένα μέταλλο θερμαίνεται καταρχήν μέχρι τήξεως και στη συνέχεια ψύχεται αργά με στόχο την επίτευξη ομοιογενών μοριακών σχηματισμών που ευνοούν τις ιδιότητες σκληρότητας του παρασκευάσματος. Κατά την ψύξη μειώνεται η ελεύθερη ενέργεια του υλικού, και τα συστατικά του σωματίδια δημιουργούν πλέγματα κρυστάλλων. Αν η ψύξη είναι αργή, δημιουργείται στην ιδανική περίπτωση ένας μόνον ομοιογενής κρύσταλλος, και η αντίστοιχη ελεύθερη ενέργεια μειώνεται στο μηδέν (απόλυτο ελάχιστο). Αν η ψύξη γίνει γρήγορα, παρουσιάζονται ανομοιογένειες και ατέλειες στην κρυσταλλοποίηση, που οδηγούν σε ένα τοπικό ελάχιστο της ελεύθερης ενέργειας.

Μια ανάλογη διαδικασία μπορεί να επιχειρηθεί κατά την αριθμητική ελαχιστοποίηση μιας συνάρτησης  $f(x)$ .

Εκκινούμε με ένα σημείο  $x(1)$  και υπολογίζουμε την αντίστοιχη τιμή της αντικειμενικής συνάρτησης  $f(x(1))$ . Δίδουμε σε μια παράμετρο θερμοκρασίας  $T$  μια αρκούντως υψηλή τιμή  $T=T_{\max}$ . Με δείκτη επαναλήψεων  $k=1$  εκτελούνται εν συνεχεία τα ακόλουθα επαναληπτικά βήματα:

(i) Στοχαστικός προσδιορισμός ενός βήματος  $\Delta x(k)$ .

(ii) Υπολογισμός  $\Delta f = f(x(k)+\Delta x(k))-f(x(k))$ . Αν  $\Delta f \leq 0$  τότε  $\Delta f = 0$ .

(iii) Προσδιορισμός του επόμενου επαναληπτικού  $x(k+1)$  με τη βοήθεια μιας γεννήτριας τυχαίων αριθμών, ως εξής:

$x(k+1) = x(k) + \Delta x(k)$  με πιθανότητα  $P(\Delta f)$

$x(k+1) = x(k)$  με πιθανότητα  $1-P(\Delta f)$

όπου  $P(\Delta f) = \exp(-\Delta f/kBT)$  και  $kB$  είναι μια σταθερή παράμετρος που αντιστοιχεί στην σταθερά Boltzmann του φυσικού φαινομένου. Αντικατάσταση  $k := k+1$ .

(iv) Αποφασίζεται αν θα μειωθεί η θερμοκρασία  $T$ . Αν όχι, πηγαινε στο (i).

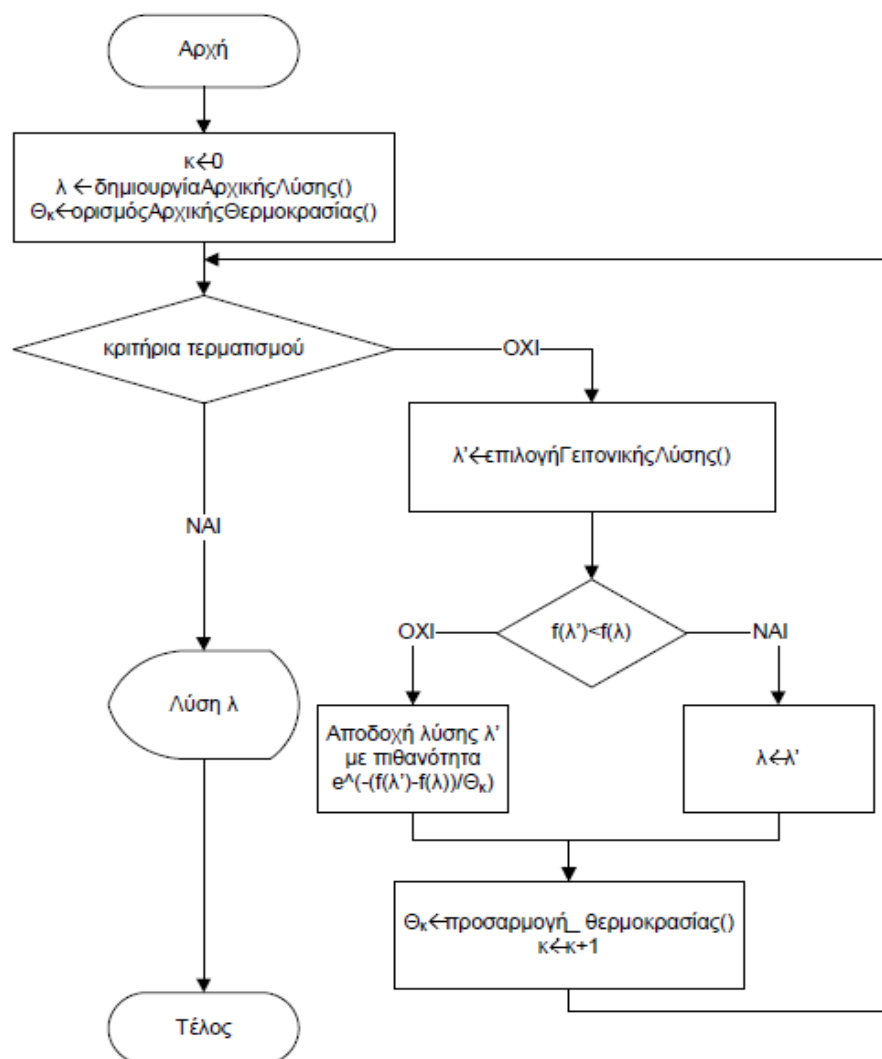
(v) Μείωση της θερμοκρασίας κατά  $\Delta T$ . Αν  $T > 0$ , πηγαινε στο (i), αλλιώς stop.

Βλέπουμε ότι υπάρχει η περίπτωση στο βήμα (iii) να επιλεχθεί με κάποια πιθανότητα ένα νέο επαναληπτικό  $x(k+1)$  που είναι χειρότερο από το προηγούμενο. Η πιθανότητα αυτή είναι τόσο μεγαλύτερη, όσο υψηλότερη η θερμοκρασία  $T$  και μικρότερη η χειροτέρευση  $\Delta f > 0$ . (Σημειώνουμε ότι, αν έχουμε βελτίωση της συνάρτησης, τότε τίθεται  $\Delta f = 0$  στο βήμα (ii) και άρα το νέο επαναληπτικό  $x(k+1)$  επιλέγεται με πιθανότητα  $P(\Delta f = 0) = 1$ ).

Επειδή στις πρώτες επαναλήψεις η θερμοκρασία  $T$  είναι υψηλή και μειώνεται σταδιακά, η περιγραφείσα διαδικασία εμπεριέχει τη δυνατότητα αποφυγής μιας παγίδευσης του αλγόριθμου σε τοπικά ελάχιστα με υψηλή τιμή της αντικειμενικής συνάρτησης και κατά συνέπεια αυξάνει την πιθανότητα προσδιορισμού ενός καλύτερου τοπικού ελάχιστου ή και του απόλυτου ελάχιστου.

Αν ετίθετο εξαρχής  $T=0$  (απότομη ψύξη του τηχθέντος μετάλλου), τότε ο αλγόριθμος θα αποδέχονταν μόνο βήματα  $\Delta x(k)$  που οδηγούν σε βελτίωση της αντικειμενικής συνάρτησης, πράγμα που θα απέκλειε τη δυνατότητα “υπερπήδησης λόφων” γύρω από ένα τοπικό ελάχιστο. Η θερμοκρασία πρέπει λοιπόν να μειώνεται στο βήμα (iv) αρκούντως αργά ώστε να δοθεί στον αλγόριθμο η ευκαιρία να διαφύγει από την περιοχή έλξεως ενός ενδεχομένως κακού τοπικού ελάχιστου. Κατά το τέλος της επαναληπτικής διαδικασίας έχει μειωθεί η θερμοκρασία αρκετά ώστε να προσδιορισθεί πλέον μέσω των βημάτων (i)-(iii) το τοπικό ελάχιστο της αντίστοιχης περιοχής έλξεως που σε πολλές περιπτώσεις μπορεί όντως να είναι το ζητούμενο απόλυτο ελάχιστο.

Παρακάτω παρουσιάζεται ένα διάγραμμα ροής για την διαδικασία που ακολουθεί η προσομοιωμένη ανόπτηση:



Εικόνα 7 Διάγραμμα ροής για Simulated Annealing.

### 3.3.2 Περιορισμένη Αναζήτηση(Tabu Search)

Η μέθοδος tabu search (TS) αποτελεί ένα ευφυή τρόπο αναζήτησης που χρησιμοποιεί δομές δεδομένων για να αποθηκεύσει πληροφορίες σχετικά με τις αποφάσεις που λαμβάνονται κατά την αναζήτηση. Οι πληροφορίες αυτές χρησιμοποιούνται για να κατευθύνουν περαιτέρω την αναζήτηση προς νέες περιοχές. Ως μέθοδος αποδίδεται στον F. Glover . Το όνομα της μεθόδου tabu ή taboo οφείλεται στο γεγονός ότι κατατάσσει ορισμένες λύσεις που προκύπτουν κατά την αναζήτηση ως απαγορευμένες επιδιώκοντας την αποφυγή εγκλωβισμού της διαδικασίας σε κάποια τοπικά βέλτιστη λύση. Θα πρέπει να σημειωθεί ότι η μέθοδος TS είναι στην βασική της μορφή μια ντετερμινιστική μέθοδος αν και έχουν υπάρξει στοχαστικές παραλλαγές της όπως η μέθοδος Probabilistic Tabu Search.

Ο αλγόριθμος TS ξεκινά με μια αρχική λύση που παράγεται είτε με τυχαίο τρόπο είτε ως αποτέλεσμα κάποιου άλλου αλγορίθμου και η οποία λαμβάνει ταυτόχρονα την θέση της τρέχουσας και της καλύτερης λύσης. Παράλληλα αρχικοποιείται και η μνήμη tabu η οποία μπορεί να αποτελείται από πολλές επιμέρους λίστες. Ο ρόλος της μνήμης δεν είναι να αποτρέψει κινήσεις που έχουν γίνει από το να επαναληφθούν αλλά να αποτρέψει την αντιστροφή τους. Για την τρέχουσα λύση καθορίζεται ένα σύνολο από γειτονικές λύσεις που προκύπτουν από αυτή μέσω κάποιου μετασχηματισμού. Οι λύσεις αυτές εξετάζονται και επιλέγεται η λύση που ελαχιστοποιεί την συνάρτηση κόστους λαμβάνοντας υπόψη την μνήμη tabu η οποία μπορεί να αποκλείει κάποιες λύσεις ή να εισάγει ποινή γι' αυτές στην συνάρτηση κόστους δυσχεραίνοντας με αυτό τον τρόπο την επιλογή τους. Η επιλεγείσα λύση αντικαθιστά την καλύτερη λύση αν έχει ευνοϊκότερη τιμή συνάρτησης κόστους από αυτή.

Σε κάθε βήμα επιλέγεται μια νέα τρέχουσα λύση ακόμα και αν δεν έχει καλύτερο κόστος σε σχέση με την τρέχουσα. Η μνήμη tabu ενημερώνεται αναλύοντας επιμέρους χαρακτηριστικά της τρέχουσας λύσης.

Τέτοια χαρακτηριστικά μπορούν να είναι ιδιότητες της λύσης που αλλάζουν κατά τον μετασχηματισμό από την παλιά τρέχουσα λύση στην νέα. Οι πληροφορίες που εισάγονται στην μνήμη tabu δεν διατηρούνται συνεχώς αλλά καθώς η διαδικασία εξελίσσεται νέες τιμές αντικαθιστούν παλιότερες. Το μέγεθος της μνήμης είναι ιδιαίτερα

σημαντικό για την απόδοση της μεθόδου ενώ έχουν προταθεί και μεταβλητά μεγέθη μνήμης. Ο αλγόριθμος TS εξετάζει με ιδιαίτερο τρόπο την περίπτωση στην οποία εντοπίζεται μια νέα λύση με καλύτερη συνάρτηση κόστους σε σχέση με όλες τις λύσεις που έχουν παρατηρηθεί αλλά η μνήμη tabu την αποκλείει. Σε αυτή την περίπτωση η λύση στην πλειονότητα των υλοποιήσεων της μεθόδου γίνεται δεκτή ενώ η διαδικασία ονομάζεται aspiration. Γενικότερα το σκεπτικό πίσω από την διαδικασία aspiration είναι ότι αν η επιστροφή σε ήδη υπάρχουσα λύση δεν μπορεί να συμβεί τότε η πληροφορία tabu μπορεί να αγνοηθεί. Συχνά στον αλγόριθμο TS και προκειμένου να ελεγχθεί αποτελεσματικότερα ο χώρος των λύσεων εφαρμόζεται η τεχνική της διαφοροποίησης. Για να γίνει αυτό μπορούν οι περιορισμοί να μεταφερθούν ως ποινή στην συνάρτηση κόστους ή να χρησιμοποιηθεί μια σημαντικά μεγαλύτερη μνήμη tabu για όσο διαρκεί η διαφοροποίηση έτσι ώστε να προκύψουν λύσεις με διαφορετικά χαρακτηριστικά από τις ήδη υπάρχουσες. Ο τερματισμός της διαδικασίας μπορεί να γίνει όταν για έναν αριθμό από επαναλήψεις δεν προκύπτει νέα καλύτερη λύση ή όταν η συνάρτηση κόστους ξεπεράσει κάποια προκαθορισμένη τιμή κατωφλίου.

Η μνήμη tabu μπορεί να διαχωριστεί σε βραχεία μνήμη, σε μνήμη συχνотήτων και σε μνήμη χρονικής εγγύτητας. Η βραχεία μνήμη (short-term memory) καταγράφει πληροφορίες σχετικά με τις λύσεις που έχει ήδη επισκεφθεί η αναζήτηση. Συνήθως καταγράφονται οι τελευταίοι μετασχηματισμοί που εφαρμόστηκαν με στόχο την αποφυγή ενεργειών που θα τους ακύρωναν. Η ποσότητα και το είδος της πληροφορίας που καταγράφεται μπορεί να διαφέρει από υλοποίηση σε υλοποίηση ενώ η διάρκεια παραμονής κάθε πληροφορίας στην μνήμη εξαρτάται από τον χρόνο. Η βραχεία μνήμη μπορεί να αποτελείται από πολλαπλές λίστες στην περίπτωση που είναι επιθυμητή η ταυτόχρονη καταγραφή διακριτών ιδιοτήτων της τρέχουσας λύσης.

Ο τυπικός τρόπος υλοποίησης μιας λίστας tabu είναι με κυκλική λίστα σταθερού μεγέθους. Ωστόσο μια λίστα με σταθερό μέγεθος έχει αποδειχθεί ότι δεν μπορεί να αποτρέψει φαινόμενα επιστροφής σε λύσεις που ήδη έχουν θεωρηθεί. Μερική λύση στο παραπάνω πρόβλημα δίνει η χρήση λιστών μεταβαλλόμενου μήκους και η τυχαία επιλογή διάρκειας αποκλεισμού (tabu tenure) για κάθε πληροφορία που εισέρχεται στην λίστα tabu. Στην μνήμη μεγάλης διάρκειας (long-

term memory) ή αλλιώς μνήμη συχνοτήτων(frequency memory) καταγράφεται ο συνολικός αριθμός επαναλήψεων στις οποίες συγκεκριμένα τμήματα της λύσης έχουν παρατηρηθεί. Προκειμένου να επιτευχθεί αποδοτικότερη εξέταση του χώρου αναζήτησης μπορεί να εφαρμοστεί η πριμοδότηση ενσωμάτωσης σπάνιων τμημάτων στην λύση. Η μνήμη χρονικής εγγύτητας (recency memory) καταγράφει τον αριθμό των συνεχόμενων επαναλήψεων στις οποίες τμήματα της λύσης παρατηρούνται χωρίς διακοπή. Σε μια προαιρετική για την μέθοδο tabu φάση εντατικοποίησης (intensification) μπορούν αυτά τα τμήματα της λύσης να παραμείνουν σταθερά και να αναζητηθεί καλύτερη λύση με τροποποίηση μόνο των εναπομεινάντων τμημάτων που σχηματίζουν την λύση.

### **Ψευδοκώδικας για Tabu Search**

***Αλγόριθμος Περιορισμένη Αναζήτηση***

***Αρχικοποίηση***

***Κατασκευή μιας αρχικής λύσης  $S_0$***

***Υπολογισμός της συνάρτησης κόστους της λύσης***

***$S^*=S_0!$  Αρχικοποίηση της βέλτιστης λύσης***

***$f(S^*)=f(S_0)$***

***Do while κάποιο κριτήριο τερματισμού δεν έχει ικανοποιηθεί***

***Υπολογισμός μιας γειτονικής λύσης  $S'$***

***If  $f(S') < f(S^*)$  then***

***$S^*=S'$***

***$f^*=f(S')$***

***end if***

***Αποθήκευσε την τελευταία κίνηση στην λίστα περιορισμένων***

***υποψηφίων(ταυτόχρονα εάν έχει συμπληρωθεί το μέγεθος της λίστας, διέγραψε την παλιότερη εγγραφή).***

***Κάλεσε κάθε  $K_1$  επαναλήψεις την στατηγική εντατικοποίησης.***



***If  $f(S_{intensification}) < f(S^*)$  then***

***$S^* = S_{intensification}$***

***$f^* = f(S_{intensification})$***

***end if***

***Κάλεσε κάθε  $K_2$  επαναλήψεις τη στρατηγική διαφοροποίησης***

***If  $f(S_{diversification}) < f(S^*)$  then***

***$S^* = S_{diversification}$***

***$f^* = f(S_{diversification})$***

***end if***

***End Do***

***Επέστρεψε τη βέλτιστη τιμή  $S^*$***

### 3.3.3 Γενετικοί Αλγόριθμοι(Genetic Algorithms)

Σε έναν τυπικό Γενετικό Αλγόριθμο, ένας πληθυσμός δομών αρχικοποιείται και κατόπιν εξελίσσεται από γενιά σε γενιά με την εφαρμογή της επιλογής, του ανασυνδυασμού και της μετάλλαξης. Το μέγεθος του πληθυσμού είναι γενικά σταθερό στους ΕΑ, αν και δεν υπάρχει συγκεκριμένος λόγος (εκτός της προγραμματιστικής ευκολίας) να μένει σταθερό.

Τυπικά, ένας εξελικτικός αλγόριθμος αρχικοποιεί τον πληθυσμό του σε τυχαίες τιμές. Ακολουθεί η αξιολόγηση (evaluation) του πληθυσμού αποδίδοντας αντίστοιχες τιμές ποιότητας (fitness) σε κάθε άτομο του πληθυσμού στο συγκεκριμένο περιβάλλον. Η αξιολόγηση γίνεται μέσω της συνάρτησης ποιότητας η οποία μπορεί να είναι πολύ απλή, όπως ο υπολογισμός μιας απλής συνάρτησης, ή εξαιρετικά πολύπλοκη, όπως η εκτέλεση μιας πολύπλοκης προσομοίωσης. Η επιλογή (selection) συνήθως υλοποιείται σε δύο βήματα, δηλ. επιλογή γονέων και επιβίωση γονέων. Κατά την επιλογή των γονέων καθορίζεται ποια άτομα θα γίνουν γονείς και πόσους απόγονους / παιδιά (offsprings/children) θα αποκτήσουν.

***procedure*EA; {**

***t = 0;***

***initialize population P(t);***

***evaluate P(t);***

***until (done) {***

***t = t + 1;***

***parent\_selection P(t);***

***recombine P(t);***

***mutate P(t);***

***evaluate P(t);***

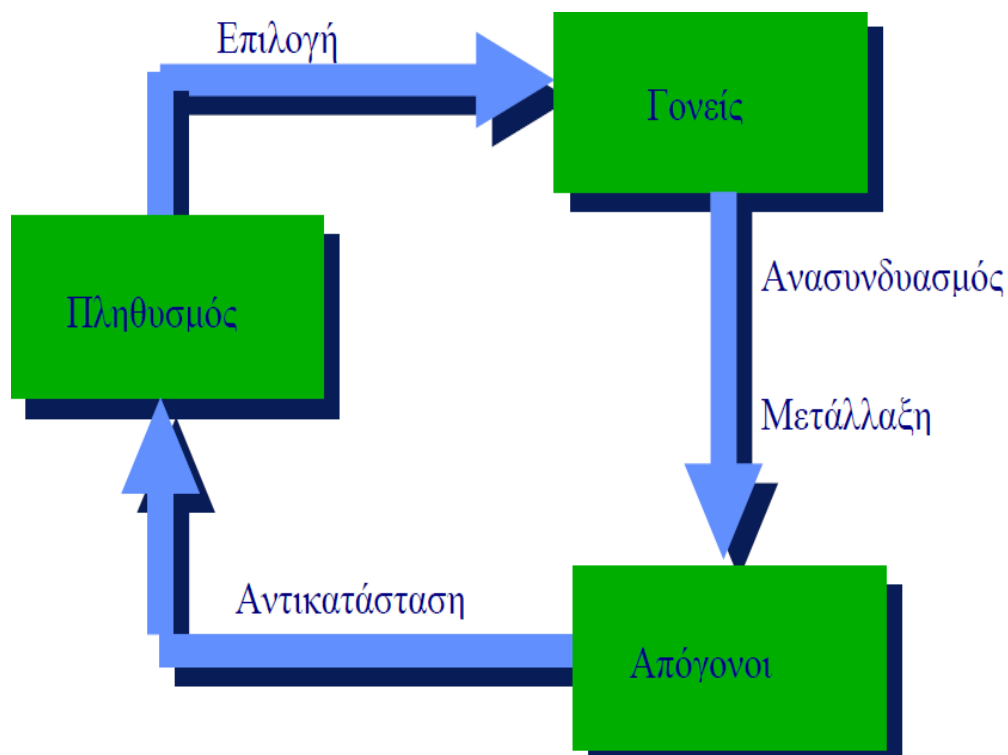
***surviveP(t);***

***}***

***}***

Οι απόγονοι δημιουργούνται μέσω ανασυνδυασμού των γονέων δηλαδή με την ανταλλαγή πληροφορίας μεταξύ των γονέων και μέσω μετάλλαξης η οποία διαταράσσει περαιτέρω τους απογόνους.

Ακολουθεί η χρήση της συνάρτησης ποιότητας για την αξιολόγηση των απογόνων και τελικά η επιλογή των ατόμων του πληθυσμού που θα επιβιώσουν στην επόμενη γενιά. Η διαδικασία αυτή ονομάζεται εξελικτικός κύκλος (evolutionary cycle). Το σχήμα παρακάτω εμφανίζει τον συνήθη εξελικτικό κύκλο.



Εικόνα 8 Εξελικτικός Κύκλος.

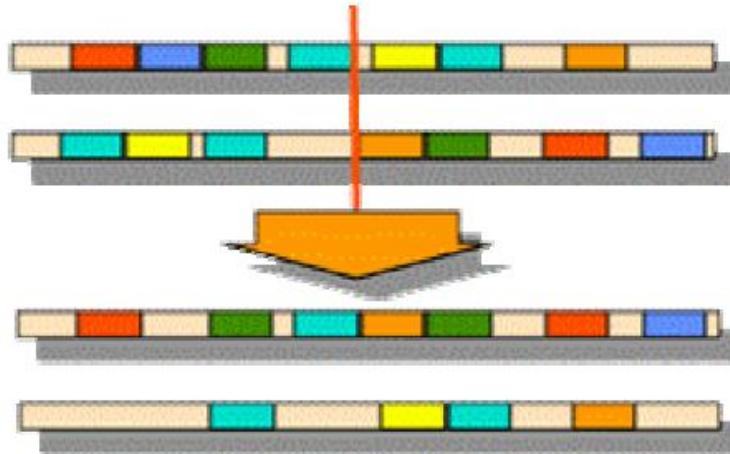
Η ανάπτυξη των αλγορίθμων αυτών ξεκίνησε στη δεκαετία του 1960 από τον John Holland, τους συνεργάτες του και τους φοιτητές του στο Πανεπιστήμιο του Michigan. Οι σκοποί της έρευνάς τους είχαν διπλή κατεύθυνση:

- να συνοψίσουν και να εξηγήσουν αυστηρά τις προσαρμοστικές και αναπαραγωγικές διαδικασίες των φυσικών συστημάτων, και
- να σχεδιάσουν λογισμικό τεχνητών συστημάτων που να διατηρεί τους πιο σημαντικούς από τους μηχανισμούς των φυσικών συστημάτων.

Παραδοσιακά, οι Γενετικοί Αλγόριθμοι (ΓΑ) χρησιμοποιούν μία αναπαράσταση η οποία είναι ανεξάρτητη από το πρόβλημα, δηλ. συμβολοσειρές δυαδικών ψηφίων. Ωστόσο, αρκετές πρόσφατες εφαρμογές χρησιμοποιούν άλλους τρόπους αναπαράστασης όπως γράφοι, εκφράσεις LISP, διατεταγμένες λίστες και διανύσματα πραγματικών αριθμών.

Κατά την αρχικοποίηση του πληθυσμού δημιουργείται ένα σύνολο από δυαδικές συμβολοσειρές (χρωμοσώματα). Μετά την αρχικοποίηση επιλέγονται οι γονείς σύμφωνα με μία συνάρτηση πιθανότητας η οποία βασίζεται στην σχετική ποιότητα των ατόμων του πληθυσμού. Με άλλα λόγια, τα άτομα (χρωμοσώματα) με καλύτερη ποιότητα έχουν περισσότερες πιθανότητες να επιλεγούν ως γονείς. Όσο καλύτερη είναι η ποιότητα ενός ατόμου, τόσο αυξάνονται οι πιθανότητες να επιλεγεί περισσότερες φορές σαν γονέας για την αναπαραγωγή παιδιών. Γενικά, από  $N$  γονείς αναπαράγονται  $N$  παιδιά μέσω διασταύρωσης (crossover), όπως ονομάζεται ο ανασυνδυασμός στην περίπτωση των ΓΑ. Τυπικά, ακολουθεί η μετάλλαξη των  $N$  παιδιών σύμφωνα με κάποιο συντελεστή πιθανότητας μετάλλαξης και η επιβίωση των παιδιών αντικαθιστώντας τους  $N$  γονείς του πληθυσμού και δημιουργώντας μία νέα γενιά.

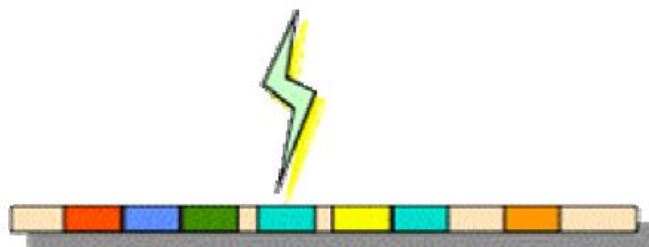
Ο τελεστής διασταύρωσης λειτουργεί ως εξής: Επιλέγεται τυχαία μία θέση του χρωμοσώματος και ανταλλάσσονται τα τμήματα πριν και μετά την θέση αυτή μεταξύ των δύο χρωμοσωμάτων όπως φαίνεται παρακάτω. Για παράδειγμα, εάν οι συμβολοσειρές 10000100 και 1111111 διασταυρωθούν μετά την τρίτη θέση θα παραχθούν τα δύο παιδιά 10011111 και 11100100.



Εικόνα 9 Διασταύρωση ενός σημείου.

Ο τελεστής διασταύρωσης μιμείται χονδρικά τον βιολογικό ανασυνδυασμό μεταξύ δύο οργανισμών με μονό χρωμόσωμα (απλοειδείς). Μέσω της διασταύρωσης, οι ΓΑ εκμεταλλεύονται αποτελεσματικά ιστορικές πληροφορίες για να κάνουν υποθέσεις πάνω σε νέα σημεία έρευνας, με προσδοκώμενη βελτιωμένη απόδοση.

Ο τελεστής μετάλλαξης αλλάζει τυχαία κάποια από τα δυαδικά ψηφία ενός χρωμοσώματος, μετατρέποντάς τα από 0 σε 1 ή αντίστροφα όπως φαίνεται στο επόμενο σχήμα. Για παράδειγμα, εάν στην συμβολοσειρά 00000100 γίνει μετάλλαξη στο δεύτερο δυαδικό ψηφίο της θα γίνει 01000100. Η μετάλλαξη μπορεί να συμβεί σε οποιαδήποτε θέση μιας συμβολοσειράς με κάποια πιθανότητα, συνήθως πολύ μικρή όσον αφορά τους ΓΑ (π.χ. 0.001).



Εικόνα 10 Μετάλλαξη.

Είναι αρκετά σημαντικό να σημειώσουμε ότι στους ΓΑ η έμφαση δίνεται στον τελεστή ανασυνδυασμού και όχι στον τελεστή μετάλλαξης. Όπως ήδη ειπώθηκε η πιθανότητα μετάλλαξης (δηλ. αντιστροφής) των

δυναμικών ψηφίων είναι πολύ μικρή και συχνά θεωρείται τελεστής που λειτουργεί στο παρασκήνιο. Ο ανασυνδυασμός, από την άλλη, θεωρείται ως ο κύριος τελεστής διερεύνησης.

Γενικά τα πλεονεκτήματα των Γενετικών αλγορίθμων, είναι:

- Μπορούν να επιλύουν δύσκολα προβλήματα γρήγορα και αξιόπιστα.
- Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα.
- Είναι εύκολα επεκτάσιμοι και εξελίξιμοι.
- Μπορούν να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους.
- Εφαρμόζονται σε πολύ περισσότερα πεδία από κάθε άλλη μέθοδο.
- Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται.
- Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας.

Γίνεται ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας.

#### 3.3.4 Αλγόριθμος Βελτιστοποίησης Αποικίας Μυρμηγκιών(ACO)

Η Βελτιστοποίηση Αποικίας Μυρμηγκιών είναι ένα σύστημα που μιμείται τη συμπεριφορά των πραγματικών μυρμηγκιών κατά τη διαδικασία της εύρεσης της τροφής τους. Τα μυρμήγκια αναπτύσσουν μια τεχνική για να βρουν τη συντομότερη διαδρομή από τη φωλιά τους προς την πηγή της τροφής τους και αντιθέτως. Τα μυρμήγκια ξεκινούν την αναζήτηση της τροφής γύρω από την πηγή με τυχαίο τρόπο και καθώς κινούνται αφήνουν πίσω τους μια ποσότητα μιας ουσίας που ονομάζεται φερομόνη και με αυτό τον τρόπο μαρκάρουν το μονοπάτι που έχουν διανύσει. Η ποσότητα της φερομόνης στο κάθε μονοπάτι

εξαρτάται από την απόσταση, την ποιότητα και την ποσότητα της τροφής που βρέθηκε. Το επόμενο μυρμήγκι που θα φύγει από τη φωλιά του είναι πολύ πιθανό να ακολουθήσει τη φερομόνη που θα υπάρχει σε κάποιο μονοπάτι, αφήνοντας μια ποσότητα φερομόνης στο ίδιο μονοπάτι. Καθώς η ποσότητα φερομόνης στο συγκεκριμένο μονοπάτι όλο και αυξάνεται όλο και περισσότερα μυρμήγκια ακολουθούν αυτό το μονοπάτι.

Όμως καθώς η ώρα περνάει η φερομόνη, ιδιαίτερα από τα μονοπάτια που δεν πηγαίνουν πολλά μυρμήγκια, ελαττώνεται. Τελικά, από όλα τα υπόλοιπα μονοπάτια η φερομόνη εξαφανίζεται και όλα τα μυρμήγκια ακολουθούν τελικά το ίδιο μονοπάτι, που είναι και η βέλτιστη ή η σχεδόν βέλτιστη λύση.

Η κύρια ιδέα της Βελτιστοποίησης Αποικίας Μυρμηγκιών είναι να μοντελοποιηθεί το πρόβλημα ως πρόβλημα εύρεσης μονοπατιού ελαχίστου κόστους σε ένα γράφημα. Κάθε μυρμήγκι αποτελεί μία λύση για το πρόβλημα. Ο αλγόριθμος αποτελείται από έναν αριθμό από επαναλήψεις, όπου σε κάθε επανάληψη κάθε μυρμήγκι αρχίζει να κατασκευάζει μια λύση με βάση την εμπειρία που έχει αποκτηθεί από τις λύσεις που δημιούργησαν τα προηγούμενα μυρμήγκια. Στο τέλος των επαναλήψεων όλα ή σχεδόν όλα τα μυρμήγκια ακολουθούν την ίδια διαδρομή η οποία κρατείται ως βέλτιστη λύση.

Η παρουσίαση της λειτουργίας του αλγορίθμου βελτιστοποίησης αποικίας μυρμηγκιών θα γίνει μέσω της παρουσίασης μιας εφαρμογής του στο πρόβλημα του πλανόδιου πωλητή που είναι το πρώτο πρόβλημα που εφαρμόστηκε η συγκεκριμένη μέθοδος και που είναι το κλασικότερο πρόβλημα αναφοράς για οποιονδήποτε αλγόριθμο που εφαρμόζεται σε προβλήματα συνδυαστικής βελτιστοποίησης. Στο πρόβλημα του πλανόδιου πωλητή, η φερομόνη συσχετίζεται με τα τόξα και για αυτό το λόγο η φερομόνη  $\tau_{ij}$  συμβολίζει την επιθυμία ο πωλητής να επισκεφτεί αμέσως μετά την πόλη  $i$  την πόλη  $j$ . Η ευρετική πληροφορία (συνάρτηση) που χρησιμοποιείται στο συγκεκριμένο πρόβλημα είναι  $\eta_{ij} = 1/c_{ij}$ , όπου  $c_{ij}$  είναι η απόσταση από το τόξο  $i$  στο τόξο  $j$  και η ευρετική συνάρτησή μας λέει ότι η επιθυμία ο πωλητής να πάει από την πόλη  $i$  στην πόλη  $j$  είναι αντιστρόφως ανάλογη με την απόσταση ανάμεσα στις πόλεις. Αν το κόστος  $c_{ij}$  είναι μηδέν τότε δίνεται στην ευρετική συνάρτηση μια πολύ μικρή τιμή. Γενικά, η ευρετική πληροφορία, αντιπροσωπεύει μια εκ των προτέρων πληροφορία για το πρόβλημα και η οποία παρέχεται από μία ανεξάρτητη πηγή από τα μυρμήγκια. Θα μπορούσε να είναι το κόστος ή μία εκτίμηση του

κόστους. Αυτή η πληροφορία, όπως θα δούμε στη συνέχεια, είναι πολύ σημαντική γιατί χρησιμοποιείται από τα μυρμήγκια για να πάρουν πιθανοτικές αποφάσεις για το πως θα κινηθούν μέσα στο γράφημα. Οι κύκλοι στο πρόβλημα του πλανόδιου πωλητή κατασκευάζονται ως εξής:

Αρχικά παίρνουμε έναν αριθμό από μυρμήγκια και τα αφήνουμε σε μία διαφορετική πόλη το καθένα (η επιλογή της πόλης γίνεται είτε τυχαία είτε πιο οργανωμένα με τη χρήση κάποιου κριτηρίου), στη συνέχεια χρησιμοποιούμε τη φερομόνη και την ευρετική συνάρτηση για την κατασκευή, με βάση τις πιθανότητες, ενός κύκλου που θα ξεκινάει και θα τελειώνει στην πόλη που αφήσαμε το μυρμήγκι. Όταν όλα τα μυρμήγκια έχουν κατασκευάσει τον κύκλο τους μπορούν να αφήσουν φερομόνη στις πόλεις που επισκέφθηκαν. Έχουν γίνει, αρκετές παραλλαγές του αρχικού αλγορίθμου στο συγκεκριμένο σημείο. Δηλαδή στο σημείο που μας λέει ποια μυρμήγκια θα αφήσουν και πόση ποσότητα φερομόνης σε ποια τόξα. Αυτό είναι ένα πολύ σημαντικό ερώτημα που τίθεται στον αλγόριθμο γιατί αν όλα τα μυρμήγκια αφήσουν την ίδια ποσότητα φερομόνης στους κύκλους που έχουν επιλέξει, όπως είναι λογικό, στην ουσία δεν θα ξεχωρίσει εύκολα και γρήγορα μια καλή διαδρομή και με αυτόν τον τρόπο δεν θα μπορέσει ο αλγόριθμος να προχωρήσει και να βρει καλύτερες λύσεις. Μια πολύ καλή στρατηγική είναι να αφήσει φερομόνη μόνο το καλύτερο μυρμήγκι (δηλαδή το μυρμήγκι με την καλύτερη τιμή στη συνάρτηση καταλληλότητας).

Πολλές φορές πριν ξεκινήσει η επόμενη επανάληψη και πριν αποφασιστεί πιο μυρμήγκι θα αφήσει φερομόνη στη διαδρομή του και τι ποσότητα φερομόνης θα αφήσει, στη λύση που βρέθηκε για κάθε μυρμήγκι εφαρμόζεται μια διαδικασία τοπικής αναζήτησης με στόχο την όσο το δυνατό βελτίωση της λύσης. Ένας αρχικός τρόπος για να υπολογιστεί η αρχική φερομόνη είναι :

$$\tau_{ij}=m/TC$$

όπου  $m$  είναι το πλήθος των μυρμηγκιών και  $TC$  είναι το κόστος ενός αρχικού κύκλου που έχει κατασκευαστεί με έναν απλούστερο τρόπο, για παράδειγμα με τη χρήση του αλγορίθμου του πλησιέστερου γείτονα. Αν η αρχική φερομόνη που δίνεται στο πρόβλημα έχει πολύ



μικρή τιμή τότε θα υπάρχει η πιθανότητα ένα μυρμήγκι που θα έχει μια σχετικά καλή αρχική λύση να κυριαρχήσει πάνω στην αναζήτηση και να οδηγήσει τον αλγόριθμο σε τοπικό βέλτιστο. Από την άλλη μεριά αν η αρχική τιμή της φερομόνης είναι πολύ μεγάλη θα πρέπει να περάσει ένας πολύ μεγάλος αριθμός από επαναλήψεις μέχρι να μπορέσει να επηρεάσει η αλλαγή της φερομόνης την λύση. Όπως είπαμε για να κατασκευαστούν οι κύκλοι των μυρμηγκιών αρχικά αφήνονται ένας αριθμός από μυρμήγκια σε διάφορες πόλεις προσέχοντας να μην είναι ο αριθμός των μυρμηγκιών μεγαλύτερος από τον αριθμό των πόλεων ώστε να μην χρειαστεί να αφήσουμε δύο μυρμήγκια πάνω στην ίδια πόλη έχοντας την πιθανότητα να δημιουργήσουμε δύο φορές τον ίδιο κύκλο. Σε κάθε βήμα κατασκευής, κάθε μυρμήγκι εφαρμόζει έναν πιθανοτικό κανόνα για να επιλέξει σε ποια πόλη θα πάει στη συνέχεια. Το κάθε μυρμήγκι που βρίσκεται σε μια πόλη  $i$  επιλέγει αν θα πάει ή όχι στην πόλη  $j$  με βάση την πιθανότητα :

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} [n_{ij}]^{\beta}}{\sum_{l=1}^M [\tau_{il}]^{\alpha} [n_{il}]^{\beta}}$$

όπου  $M$  είναι το σύνολο των πόλεων και  $\alpha, \beta$  δύο παράμετροι που καθορίζουν αν επηρεάζει περισσότερο την επιλογή η φερομόνη που βρίσκεται πάνω στο τόξο ή η ευρετική πληροφορία που έχει υπολογιστεί για κάθε τόξο. Με αυτόν τον πιθανοτικό κανόνα η πιθανότητα επιλογής ενός τόξου αυξάνει όταν αυξάνει η φερομόνη του τόξου και η ευρετική πληροφορία του τόξου. Αν το  $\alpha = 0$  οι κοντινότερες πόλεις είναι πιο πιθανό να επιλεγούν, κάτι που μετατρέπει τον αλγόριθμο σε κλασικό αλγόριθμο του πλησιέστερου γείτονα και η παραπάνω εξίσωση γίνεται :

$$p_{ij} = \frac{[n_{ij}]^\beta}{\sum_{l=1}^M [n_{il}]^\beta}$$

Εάν το  $\beta = 0$  τότε μόνο η φερομόνη χρησιμοποιείται πράγμα που έχει δείξει ότι δεν οδηγεί σε πολύ καλά αποτελέσματα γιατί ο αλγόριθμος δεν λαμβάνει υπόψη του καθόλου τις αποστάσεις παρά μονάχα τη φερομόνη πράγμα που μετατρέπει τον αλγόριθμο σε καθαρά πιθανοτικό αλγόριθμο με τυχαία βήματα και η Εξίσωση γίνεται :

$$p_{ij} = \frac{[\tau_{ij}]^\alpha}{\sum_{l=1}^M [\tau_{il}]^\alpha}$$

Όταν όλα τα μυρμήγκια έχουν κατασκευάσει τους αρχικούς κύκλους τους, η ποσότητα φερομόνης που υπάρχει σε κάθε τόξο ενημερώνεται και μια καινούργια ποσότητα φερομόνης προστίθεται στα τόξα που χρησιμοποιήθηκαν από τα μυρμήγκια. Αρχικά, η φερομόνη μειώνεται σε όλα τα τόξα κατά μία ποσότητα :

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad (4)$$

όπου  $0 < \rho < 1$  είναι ένα συντελεστής εξάτμισης και χρησιμοποιείται για να εξασφαλίσει ότι ο αλγόριθμος θα ξεχάσει κακές αποφάσεις που έχουν γίνει σε προηγούμενες επαναλήψεις. Έτσι, αν ένα τόξο δεν επιλέγεται από τα μυρμήγκια τότε η φερομόνη του τόξου μειώνεται εκθετικά με τον αριθμό των επαναλήψεων μέχρι να πλησιάσει κοντά

στο μηδέν. Στη συνέχεια, η φερομόνη των τόξων που χρησιμοποιήθηκαν από τα μυρμήγκια αυξάνεται κατά:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k$$

όπου ο όρος του αθροίσματος είναι η ποσότητα φερομόνης που αφήνει κάθε μυρμήγκι,  $k$ , στα τόξα που πέρασε και δίνεται από τον τύπο:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C^k}, & \text{εάν το τόξο έχει επιλεγεί από το } k \\ 0, & \text{αλλιώς} \end{cases}$$

όπου το  $C^k$  είναι το κόστος του κύκλου που έχει δημιουργήσει το μυρμήγκι  $k$ . Όσο καλύτερος είναι ο κύκλος που έχει δημιουργηθεί τόσο περισσότερη φερομόνη αφήνουν στα τόξα και άρα έχει περισσότερες πιθανότητες να επιλεγεί στην επόμενη επανάληψη. Υπάρχει πιθανότητα σε κάποια προβλήματα η επιλογή να αφήσουν όλα τα μυρμήγκια φερομόνη στα τόξα να οδηγήσει σε πολύ αργή σύγκλιση του αλγορίθμου. Για αυτό το λόγο έχουν εμφανιστεί αρκετές παραλλαγές της διαδικασίας ενημέρωσης της φερομόνης. έτσι, θα μπορούσε να αφήσει μόνο το βέλτιστο μυρμήγκι φερομόνη ή θα μπορούσε το μυρμήγκι με τη βέλτιστη λύση να αφήσει μια επιπλέον ποσότητα εκτός από την ποσότητα που θα αφήσουν όλα τα μυρμήγκια. Στη συνέχεια δίνεται ένας γενικός αλγόριθμος με μορφή ψευδοκώδικα του αλγορίθμου βελτιστοποίησης αποικίας μυρμηγκιών.

### **Αλγόριθμος Βελτιστοποίησης αποικίας μυρμηγκιών**

Αρχικοποίηση

Δημιουργία του αρχικού πληθυσμού των μυρμηγκιών

Υπολογισμός της ευρετικής συνάρτησης  $n_{ij}$  για κάθε τόξο του γραφήματος

Υπολογισμός της αρχικής φερομόνης  $\tau_{ij}$  για κάθε τόξο του γραφήματος

Επιλογή του μέγιστου αριθμού επαναλήψεων

*Κύρια Φάση*

**Do until** δεν έχει φθάσει ο μέγιστος αριθμός των επαναλήψεων  
Κάθε μυρμήγκι ξεκινάει τη δική του διαδρομή από διαφορετικό σημείο στο χώρο

**For** κάθε μυρμήγκι στον πληθυσμό **do**

**Do while** τα κριτήρια τερματισμού δεν ικανοποιούνται Επίλεξε το επόμενο σημείο στο γράφημα που θα πάει το μυρμήγκι βάσει της φερομόνης και της ευρετικής συνάρτησης

Υπολόγισε τη συνάρτηση ποιότητας για κάθε

Μυρμήγκι

Εφάρμοσε μία διαδικασία τοπικής αναζήτησης

**End do**

**End For**

Ενημέρωσε τη φερομόνη βάσει της λύσης του βέλτιστου μυρμηγκιού στον πληθυσμό

**End do**

Επέστρεψε το βέλτιστο μυρμήγκι (βέλτιστη λύση).

## Κεφάλαιο 4: Περιγραφή Αλγορίθμου

Για την ανάπτυξη του αλγορίθμου έγινε ένας συνδυασμός αλγορίθμων τοπικής αναζήτησης και του αλγορίθμου ACO(Ant Colony Optimization).

### 4.1: Διάβασμα μεταβλητών, αρχικοποιήσεις.

Η γενική ιδέα είναι διάβασμα δεδομένων από αρχείο excel για την αρχικοποίηση του προβλήματος, τα οποία είναι τα εξής.

**X-YCOORD:** ως οι συντεταγμένες x,y του κάθε κόμβου .

**DELIVERY:** η ποσότητα προϊόντος που παραδίδεται σε κάθε κόμβο.

**PICKUP:** η ποσότητα προϊόντος που παραλαμβάνεται από κάθε κόμβο.

**BEGINNING-ENDING TW:** ως το χρονικό παράθυρο που ο κάθε κόμβος είναι επισκέψιμος.

**SERVICE-TIME:** Ο χρόνος που θα ξοδέψει το εκάστοτε όχημα για όταν επισκεφθεί ένα κόμβο. Για λόγους ευκολίας σε κάθε παράδειγμα που θα τρέξουμε στη συνέχεια , ο αριθμός αυτός είναι ίδιος για όλους τους κόμβους , χωρίς όμως να αποκλείεται αυτό να αλλάξει , μη επηρεάζοντας τον αλγόριθμο φυσικά.

**X-YCOORDTIME:** Σε περίπτωση που θέλουμε να χρησιμοποιήσουμε διαφορετικές συντεταγμένες για την χρονική απόσταση μεταξύ 2 κόμβων. Σε πειραματικό επίπεδο μπορεί να χρησιμοποιηθεί η απόσταση σε km με προσθήκη του αντίστοιχου Service Time.

**CAPACITY:** ως η μέγιστη χωρητικότητα του οχήματος –οχημάτων.

**MAXTOURTIME:** για την μέγιστη διάρκεια διαδρομής κάθε οχήματος ( βάρδια ουσιαστικά).

**NODES:** Ο αριθμός κόμβων σε κάθε παράδειγμα.

Αξίζει να αναφερθεί επίσης ότι δηλώνεται 2διάστατος πίνακας Route για να εξυπηρετήσει την αρχική μας λύση, με πρώτο στοιχείο πάντα το 1(depot).] και επιπλέον κενός πίνακας **DistanceTable** με στοιχεία τον αριθμό των κόμβων<sup>2</sup>.

Είναι φυσικά απαραίτητο να έχουμε υπολογίσει και τις αποστάσεις αυτού του πίνακα ώστε κάθε στιγμή να γνωρίζουμε την απόσταση από

τον ένα κόμβο στον άλλο, οπότε και κάνουμε χρήση των συντεταγμένων χυτου κάθε κόμβου με εργαλείο μας το Πυθαγόρειο θεώρημα.

#### Ευκλείδεια απόσταση σε διάφορα συστήματα συντεταγμένων

Ο τύπος της απόστασης στο καρτεσιανό σύστημα συντεταγμένων βασίζεται στο πυθαγόρειο θεώρημα. Αν  $(x_1, y_1)$  και  $(x_2, y_2)$  είναι σημεία του επιπέδου, τότε η απόσταση μεταξύ τους, που ονομάζεται και ευκλείδεια απόσταση, δίνεται από τον τύπο:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

Πιο γενικά, στους ευκλείδειους  $n$ -διάστατους χώρους, η απόσταση μεταξύ 2 σημείων  $A=(\alpha_1, \alpha_2, \dots, \alpha_n)$  και  $B=(\beta_1, \beta_2, \dots, \beta_n)$ , ορίζεται με γενίκευση του Πυθαγορείου θεωρήματος ως:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}.$$

Συνεπώς έχουμε:

*For i από 1 until αριθμό κόμβων*

*For j από 1 until αριθμό κόμβων*

*DistanceTable[i,j]=Square\_Root[(x<sub>i</sub>-x<sub>j</sub>)<sup>2</sup>+(y<sub>i</sub>-y<sub>j</sub>)<sup>2</sup>]*

*End for*

*End for*

Έτσι έχουμε έτοιμο τον πίνακα αποστάσεων ο οποίος φυσικά έχει μηδενικά τα στοιχεία της κυρίας διαγωνίου του.

Για την συνέχεια που περιλαμβάνει χρήση αλγορίθμου πλησιέστερου γείτονα, αρχικοποιούμε μια μεταβλητή **position**, η οποία θα μας υποδεικνύει τον κόμβο στον οποίο βρισκόμαστε αλλά και μια επαρκώς μεγάλη τιμή(**Closest\_Distance**) για την εκκίνηση του εν λόγω αλγορίθμου.

## 4.2: Παραγωγή αρχικού αποτελέσματος με αλγόριθμο πλησιέστερου γείτονα.

Πριν ξεκινήσουμε την διαδικασία, προχωράμε στην δημιουργία ενός μονοδιάστατου πίνακα με στοιχεία όσοι και οι κόμβοι. Ο εν λόγω **Visited\_Nodes** εξυπηρετεί ανάγκη τερματισμού αλλά και ορθού αποτελέσματος καθώς όλα του τα στοιχεία αρχικά είναι μηδενικά. Έτσι κάθε φορά που καταλήγει το όχημα μας σε ένα κόμβο, το αντίστοιχο στοιχείο γίνεται 1. Παρακάτω θα δούμε πώς αυτό βοηθάει στους περιορισμούς, πάντως για τον σκοπό του τερματισμού μας ενδιαφέρει το άθροισμα των στοιχείων του να είναι όσοι και οι κόμβοι, κάτι που σημαίνει ότι τους έχουμε επισκεφτεί όλους. Συνεπώς έχουμε τα εξής:

*While Sum(VisitedNodes)not\_equal(Αριθμός Κόμβων)*

*Closest\_Distance=10000;*

*i=1;*

*Time=0;*

*for j=2 εώς Αριθμός Κόμβων*

*if Απόσταση i→j < Closest\_Distance*

*Εξίσωση Closest\_Distance με απόσταση i→j*

*Position=j*

*End\_if*

*End\_for*

Στην πρώτη επανάληψη είναι προφανές ότι ο οποιοσδήποτε κόμβος μπορεί να πάρει την θέση **position** , κάτι όμως που δεν ισχύει από εκεί και μετά. Πλέον έχουμε το στοιχείο **Position**(επόμενος κόμβος), ο οποίος είναι προς εξέταση πληρότητας των περιορισμών μας. Έτσι έχουμε στην συνέχεια:

*If 1) Time+ Απόσταση i→position + Service time(position)+ Επιστροφή  
i←position < DayTime*

2)  $Capacity > Delivery(position)$

3)  $Visited\_Nodes(position) = 0$ ; (επισκέψιμος)

Then Εκχώρηση της  $position$  στον πίνακα  $Route$  στην επόμενη θέση

Αφαίρεση του αντίστοιχου  $Delivery$  από το  $capacity$

Προσθήκη του χρόνου στο  $Time$

$Visited\_Nodes(position) = 1$ ; (μη επισκέψιμος πλέον)

$i = position$ ;

else if  $Capacity < Delivery(position)$

επιστροφή στην  $Depot$  και αρχικοποίηση όλων των μεταβλητών πλην του  $Time$ .

$i = 1$ ;

else if  $Time + Απόσταση\ i \rightarrow position + Service\ time(position) + Επιστροφή\ i \leftarrow position > DayTime$

επιστροφή στην  $Depot$  και αρχικοποίηση όλων των μεταβλητών με  $Time = 0$ ; (λήξη βάρδιας)

$i = 1$ ;

end\_if

Τερματισμός

If  $sum(Visited\_Nodes) = Αριθμός\ κόμβων - 1$

$Visited\_Nodes(Αποθήκης) = 1$

Εκχώρηση της αποθήκης ως τελευταίο στοιχείο στον  $Route$

End\_if

End\_While

Υπολογισμός κόστους διαδρομής από τον πίνακα  $Route$  με χρήση  $Distance\ Table$ .



Αξίζει να σημειωθεί πως ο έλεγχος του **Time** ως περιορισμού, μπορεί να γίνεται επιπλέον, πριν από τον έλεγχο των υπολοίπων ως εξής:

*if Time+ Απόσταση  $i \rightarrow position$  + Service time(position)+ Επιστροφή  $i \leftarrow position < DayTime$  AND position=depot.*

*Time=0;*

Έτσι αποφεύγουμε την περίπτωση να προστεθούν 2 συνεχόμενες αποθήκες στην διαδρομή μας εφόσον επιλεγθεί κόμβος που δεν μας φτάνει ο χρόνος εξυπηρέτησης και επιστροφής. Αυτό γίνεται όταν επιστρέψουμε αποθήκη με σχετικά μικρό υπολειπόμενο χρόνο.

#### 4.3: Αρχικοποίηση μεταβλητών για αλγόριθμο ACO και ο αλγόριθμος επιλογής Ρουλέτας(Roulette Wheel Selection).

Η γενικότερη δομή του προγράμματος ακολουθεί την εξής μορφή:

##### Δηλώσεις 1

Εξωτερική επανάληψη

##### Δηλώσεις 2

Επαναλήψεις μυρμηγκιών (ίσες με τον αριθμό των κόμβων)

##### Δηλώσεις 3

Τέλος\_επαναλήψεων μυρμηγκιών

Τέλος εξωτερικής επανάληψης.

## Δηλώσεις 1

Αρχικά, ξεκινάμε εκχωρώντας στην μεταβλητή **Ants**, τον αριθμό των κόμβων. Όσο περισσότερα μυρμήγκια, τόσο το καλύτερο, παράλληλα όμως δεν πρέπει να υπερβαίνουν τον αριθμό των κόμβων διότι δημιουργούνται προβλήματα αργότερα στο μέγεθος των πινάκων όπου θα αποθηκεύουμε τις διαδρομές, με πιο συχνό πρόβλημα, το index να ξεπερνάει το μέγεθος του πίνακα.

Επίσης είναι απαραίτητο να έχουμε στην διάθεσή μας 2 ακόμα πίνακες:

**NFunction**, πίνακας άσων (κόμβοι-κόμβοι)

**NTable**----→NFunction/DistanceTable για να καταλήξουμε στην Ευρετική πληροφορία  $1/C_{ij}$  για κάθε τόξο. Φυσικά λόγω της διαίρεσης με το 0 η κύρια διαγώνιος έχει τιμή inf(άπειρο), οπότε δίνουμε εμείς στα στοιχεία της μία πολύ μικρή τιμή.

Έπειτα έχουμε και πίνακα Φερομόνης (**FeromoneTable**) με κάθε στοιχείο του να είναι: **1/InitialCost** που βγάλαμε στο προηγούμενο υποκεφάλαιο. Οι αυξομειώσεις των τιμών των στοιχείων του κατά το τρέξιμο του αλγορίθμου, θα παίξει πολύ σημαντικό ρόλο για την επιλογή του κάθε κόμβου όσο αυξάνονται οι επαναλήψεις.

Στη συνέχεια χρειαζόμαστε και τον πίνακα **PTable** που αναπαριστά τον πιθανοτικό πίνακα μέσω του οποίου το κάθε μυρμήγκι επιλέγει τον επόμενο σταθμό του. Επιπλέον συντελεστής  $0 < p < 1$  ως συντελεστής εξάτμισης της φερομόνης στο τέλος της επανάληψης και πίνακας **RouteBestAnts** τύπου cell ώστε σε κάθε στοιχείο του να αποθηκεύεται η καλύτερη διαδρομή κάθε επανάληψης και το κόστος της (**BestOfIterations**).

Πριν συνεχίσουμε, να τονίσουμε ότι ο πίνακας **RouteBestAnts** αλλά και όλοι οι τύπου cell πίνακες στην συνέχεια εξυπηρετούν τον σκοπό αποθήκευσης δεδομένων όπου ο αριθμός των γραμμών είναι σταθερός, ενώ ο αντίστοιχος των στηλών όχι.

## Εξωτερική επανάληψη

Αρχικοποίηση φερομόνης με προσθήκη της τιμής κάθε στοιχείου σε περίπτωση που ο **FeromoneTable** καταλήξει να έχει πολύ μικρές τιμές.

## Δηλώσεις 2

Cell πίνακας **RouteAnts** ο οποίος γεμίζει σε κάθε εσωτερική επανάληψη με την διαδρομή κάθε μυρμηγκιού.

**AntsCostTable**, το κόστος της κάθε παραπάνω διαδρομής.

Εσωτερική επανάληψη μυρμηγκιών

## Δηλώσεις 3

Αρχικοποίηση πίνακα **PTable** ο οποίος πρέπει να είναι διαφορετικός για κάθε μυρμηγκι που αναπαριστά 1 επανάληψη του εσωτερικού βρόγχου.

Συντελεστές **a(εκθέτης φερομόνης)** και **b(εκθέτης αποστάσεων)**, για τον πιθανοτικό πίνακα.

Αρχικοποίηση capacity και πάλι , **Time** και πίνακας **VisitedNodes** όπως και πριν την εκτέλεση του αλγορίθμου πλησιέστερου γείτονα.

Εν συνεχεία έχουμε:

*While Sum(VisitedNodes)<Ants*

*If Sum(VisitedNodes)=0;*

*i=1;*

*end\_if*

*for j=1 έως nodes*

*Υπολογισμός πίνακα PTable για τον εκάστοτε κόμβο i.\**

*End\_for*

*Εφαρμογή αλγορίθμου Roulette Wheel Selection\*\**

*NextNode=Αποτέλεσμα Roulette Wheel Selection.*

*If Constraints(Capacity, Time, Visitable node)are met*

*RouteAnts(Next)=NextNode;*

*Capacity=Capacity-Capacity(NextNode)*

```

    Time=Time+Service_Time(NextNode)+Distance_to_NextNode;
    VisitedNodes(NextNode)=1;
    i=NextNode;
else
    Επιστροφή αποθήκη
    Αρχικοποίηση μεταβλητών
    i=1;
end_if
end_While

```

Σε ότι αφορά τα παραπάνω:

#### **\*Υπολογισμός πίνακα PTable**

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} [n_{ij}]^{\beta}}{\sum_{l=1}^M [\tau_{il}]^{\alpha} [n_{il}]^{\beta}}$$

#### **\*\*Αλγόριθμος Roulette Wheel Selection**

Πρόκειται για μια διαδικασία επιλογής γονέων σε γενετικούς αλγορίθμους κυρίως, που γίνεται με τη χρήση του κανόνα της Ρουλέτας (**Roulette wheel selection**). Η επιλογή δεν είναι τυχαία αλλά όσο καλύτερη είναι η ποιότητα κάποιου τόσο μεγαλύτερη είναι και η πιθανότητα αυτός ο γονέας να επιλεγεί. Αρχικά, σε μια μεταβλητή προστίθενται οι τιμές όλων των συναρτήσεων ποιότητας των γονέων. Στη συνέχεια, με μια γεννήτρια τυχαίων αριθμών επιλέγεται ένας αριθμός από το 1 ως το συνολικό άθροισμα της μεταβλητής. Ανάλογα

με το νούμερο που θα έχει βγει από τη γεννήτρια επιλέγεται και ο ένας από τους δύο γονείς.

Έστω ότι σε ένα πρόβλημα μεγιστοποίησης έχουμε τέσσερις υποψήφιους γονείς, όπου η συνάρτηση ποιότητας για τον κάθε γονέα έχει τιμή 95, 4, 28, 73, αντίστοιχα. Το συνολικό άθροισμα αυτών των αριθμών είναι 200. Άρα η γεννήτρια τυχαίων αριθμών θα μας δώσει μια τιμή από το ένα ως το 200. Αν η τιμή είναι από το 1 ως το 95 παίρνουμε ως γονέα τον πρώτο, αν είναι από το 96 ως το 99, παίρνουμε το δεύτερο, από 100 ως 127 τον τρίτο και αν είναι από 128 ως 200 παίρνουμε τον τέταρτο. Συνεπώς η πιθανότητα του να επιλεγεί το καλύτερο άτομο ως γονέας είναι πολύ μεγαλύτερη από το να επιλεγεί το χειρότερο άτομο. Φυσικά, αν το πρόβλημα είναι πρόβλημα ελαχιστοποίησης η διαδικασία δεν διαφέρει κατά πολύ.

Πιο συγκεκριμένα, έχουμε στον παραπάνω αλγόριθμο:

- Παραγωγή τυχαίου αριθμού **10<sup>7</sup><Randomizer<SumRoulette**(Άθροισμα στοιχείων της γραμμής του τρέχοντος κόμβου).

- Επιλογή από τον πίνακα **PTable** γραμμή **i** της πλησιέστερης στον **Randomizer** πιθανότητας και συνεπώς κόμβου.

- Εκχώρηση στο **NextNode**.

#### 4.4: Διεργασίες μετά την ολοκλήρωση διαδρομών μυρμηγκιών.

Έχοντας καταλήξει στο τέλος κάθε διαδρομής της ομάδας μυρμηγκιών, έχουμε τον πίνακα **RouteAnts(Nodes,1,\*),** όπου \* είναι το μήκος της διαδρομής κάθε μυρμηγκιού.

Γίνεται χρήση της εντολής:

*filledCells = ~cellfun(@isempty,RouteAnts);*

η οποία μας επιστρέφει πίνακα **filledCells** ίδιων διατάσεων με τον **RouteAnts** με στοιχεία 1 για κελιά που δεν είναι κενά και 0 για τα κενά. Μπορούμε συνεπώς να ξέρουμε το μήκος κάθε διαδρομής και μέσω αυτού να υπολογίσουμε το εκάστοτε κόστος. Αυτό μπορεί πλέον να μας βοηθήσει ως εξής:

*for i=1 έως AntsNumber*

*Αρχικοποίηση μετρητή κελιών*

*For j=1 έως length(RouteAnts)*

*Προσθήκη του στοιχείου filledCells(i,1,j) στο μετρητή κελιών.*

*end\_for*

*for k=1 έως μετρητής κελιών-1*

*Εύρεση συνολικού κόστους διαδρομής i από RouteAnts*

*AntsCostTable(i)=Κόστος διαδρομής i*

*end\_for*

*end\_for*

*Best εσωτερικού βρόγχου=Minimum AntsCostTable*

*BestOfIterations(next)← Best εσωτερικού βρόγχου*

*Εύρεση Best Ant κόστος εσωτερικού βρόγχου*

*Μείωση φερομόνης FeromoneTable(i,j) \* (1-p) για όλα τα i,j*

*RouteBestAnts← διαδρομή Best Ant*

*For κάθε i,j τόξο του RouteBestAnts*

*Αύξηση φερομόνης FeromoneTable(i,j)+(1/κόστος Best Ant)*

*end\_for*

*end\_for εξωτερικού βρόγχου.*

#### 4.5: Διεργασίες μετά την ολοκλήρωση του εξωτερικού βρόγχου και επεξεργασία τελικής διαδρομής.

Συνολικά μέχρι τώρα έχουμε η διαδρομές( όσες και οι επαναλήψεις του εξωτερικού βρόγχου) η οποίες επιλέχτηκαν ως οι καλύτερες με το μικρότερο κόστος μεταξύ των υπολοίπων διαδρομών σε κάθε εκτέλεση του εσωτερικού βρόγχου. Φυσικά επιδέχεται βελτίωση σε κάποιες ακόμα και το συγκεκριμένο αποτέλεσμα, κάτι που γίνεται όχι πάντα επιτυχημένα με χρήση αλγορίθμων τοπικής αναζήτησης.

Ξεκινάμε εκχωρώντας  $\text{RouteBestAnts2} \leftarrow \text{RouteBestAnts}$ .

Ο βασικός λόγος είναι οι αλλαγές που θα επέλθουν να μην επηρεάζουν τον αρχικό πίνακα σε περίπτωση που δεν έχουμε κάποια βελτίωση.

*For*  $i=1$  έως Αριθμός επαναλήψεων εξωτερικού βρόγχου

*For*  $j=1$  έως Αριθμός εκτέλεσης *LocalSearch*

*Εφαρμογή 1-1 Exchange στη διαδρομή*  $\text{RouteBestAnts2}(i)$

*Εφαρμογή 2-0 Relocate στη διαδρομή*  $\text{RouteBestAnts2}(i)$

$\text{CostBefore} \leftarrow \text{RouteBestAnts}(i)$

$\text{CostAfter} \leftarrow \text{RouteBestAnts2}(i)$

*If* *Constraints ok AND*  $\text{CostBefore} > \text{CostAfter}$

$\text{RouteBestAnts}(i) \leftarrow \text{RouteBestAnts2}(i)$

$\text{BestOfIterations}(i) \leftarrow \text{CostAfter}$

$j = \text{Αριθμός εκτέλεσης LocalSearch για τερματισμό}$

*else*  $\text{RouteBestAnts2}(i) \leftarrow \text{RouteBestAnts}(i)$

*end\_if*

*end\_for*

*end\_for*

Μετά την εκτέλεση των παραπάνω, ο πίνακας **RouteBestAnts** έχει πλέον τις διαδρομές με το καλύτερο δυνατό κόστος, είτε έχουν τροποποιηθεί είτε όχι.

*FinalCost* ← ————— *min του BestOfIterations*

*FinalRoute* ← ————— *Η διαδρομή με το FinalCost*

Το μόνο πλέον που απομένει είναι να επεξεργαστούμε την τελική διαδρομή ώστε κάθε επιστροφή στην αποθήκη που γίνεται λόγω υπέρβασης του χρονικού ορίου που έχει τεθεί, να αντιπροσωπεύει μια βάρδια οδηγού. Έτσι μειώνεται και ο συνολικός χρόνος εκτέλεση της αποστολής-παραλαβής δεμάτων.



## Κεφάλαιο 5: Αποτελέσματα και συμπεράσματα.

Συνολικά η πειραματική διαδικασία περιλαμβάνει το τρέξιμο 14 παραδειγμάτων μικρομεσαίου μεγέθους , με αριθμό κόμβων Nodes[51-200], χωρητικότητα φορτηγών Capacity[140-200]. Τα μισά από αυτά έχουν μέγιστο χρόνο διαδρομής [160-1040] με Service Time ανά κόμβο [10-90], ενώ τα υπόλοιπα 7 δεν έχουν μέγιστο χρόνο διαδρομής και συνεπώς ούτε Service Time.

Στη συνέχεια ακολουθούν παραδείγματα μεγαλύτερου μεγέθους, με αριθμό κόμβων Nodes [240-483], χωρητικότητα φορτηγών Capacity[200-1000] και μέγιστο χρόνο διαδρομής[650-1800]. Σε κανένα από αυτά τα τελευταία δεν υπάρχει Service Time και οι συντεταγμένες περιλαμβάνουν αρνητικούς αλλά και δεκαδικούς αριθμούς.

### Συμπεράσματα σχετικά με την ρύθμιση παραμέτρων.

Οι παράμετροι που επιδέχονται ρύθμισης είναι οι εξής:

Αριθμός μυρμηγκιών ,**Ants**

Αριθμός εξωτερικών επαναλήψεων ,**ExternalCounter**

Παράμετροι  **$\alpha$**  και  **$\beta$**

Συντελεστής εξάτμισης φερομόνης , **$0 < \rho < 1$**

Ίχνος φερομόνης  **$0 < q < 1$**  κατά την αύξηση φερομόνης του μονοπατιού του βέλτιστου μυρμηγκιού κατά  **$q/\text{BestAntCost}$**  στο τέλος κάθε εξωτερικού βρόγχου ExternalCounter.

**Ants:** Αντίθετα με το πρόβλημα του πλανόδιου πωλητή ,όπου προτείνεται η χρήση αριθμού μυρμηγκιών ίσο με τον αριθμό των κόμβων, τα πράγματα εδώ είναι διαφορετικά. Σε κάθε παράδειγμα χρησιμοποιήθηκε μικρός σχετικά πάντα αριθμός ants. Σε γενικές γραμμές συνδέεται με τον αριθμό των εξωτερικών επαναλήψεων και δεν φαίνεται όσο μεγαλώνει αυτός ο αριθμός να προσφέρουν κάτι παραπάνω τα επιπλέον μυρμήγκια. Αντιθέτως αυξάνεται η πιθανότητα να επιλεγεί μυρμήγκι που θα μας οδηγήσει σε πορεία μακριά από το

βέλτιστο και συνεπώς σε ανάγκη περισσότερων επαναλήψεων μέχρι την επαναφορά του κόστους σε αποδεκτά επίπεδα.

**ExternalCounter:** Αφορά τον αριθμό των εξωτερικών επαναλήψεων. Με λίγα λόγια αναφέρεται στο πόσες φορές θα γίνει η διαδικασία Τρέξιμο Ομάδας Μυρμηγκιών-Εξαγωγή Βέλτιστου-Εξάτμιση Φερομόνης-Αύξηση φερομόνης στο Βέλτιστο Μονοπάτι. Ο αριθμός των επαναλήψεων που χρησιμοποιήθηκε , είναι αρκετά μεγάλος [ $>1000$ ].

**Παράμετροι  $\alpha, \beta$ :** Οι τιμές που χρησιμοποιήθηκαν ήταν οι εξής,

$\alpha[0,0.5,1,2,5]$

$\beta[0,1,2,3,4,5]$

Μέσω των εκτελέσεων και στο συγκεκριμένο set προβλημάτων , έγινε αντιληπτό ότι για τις πολύ μικρές τιμές της παραμέτρου  $\alpha$ , το σύστημα γίνεται νομοτελειακό χωρίς μνήμη και τελικά είναι ανίκανο να βρει μια κατάλληλη λύση, επειδή δεν είναι ικανό για να συγκλίνει σε μια βέλτιστη διαδρομή. Από την άλλη μεριά, για πολύ μεγάλες τιμές στο ζεύγος  $(\alpha, \beta)$ , το σύστημα εμφανίζεται να λιμνάζει σε λύσεις που απέχουν από τη βέλτιστη λύση. Μεγάλες τιμές του  $\beta$  και μόνο έχουν παρόμοιο αποτέλεσμα, με μόνο πλεονέκτημα την γρήγορη σύγκλιση σε οριακά μη αποδεκτές τιμές.

**Συντελεστής  $\rho$ :** Όπως αναφέραμε το  $\rho$  αποτελεί τον συντελεστή εξάτμισης της φερομόνης. Λόγω του πλήθους των εξωτερικών επαναλήψεων που χρησιμοποιούμε, οι τιμές κυμαίνονται για  $\rho[0.01, 0.1]$

**Ίχνος  $q$ :** Εδώ οι τιμές για το  $q$  ήταν  $[0.25, 1]$  με βήμα 0.25. Παρόλα αυτά δεν παρατηρήθηκε κάποια συγκλονιστική βελτίωση ή επιδείνωση με την διαφοροποίησή τους.

Εν συνεχεία παρουσιάζονται τα παραδείγματα και τα αποτελέσματα για τους (par1-par14):

PAR	Nodes	Capacity	Route_Time	Service_Time
1	51	160	infinite	0
6	51	160	200	10
2	76	140	infinite	0
7	76	140	160	10
3	101	200	infinite	0
8	101	200	230	10
12	101	200	infinite	0
14	101	200	1040	90
11	121	200	infinite	0
13	121	200	720	50
4	151	200	infinite	0
9	151	200	200	10
5	200	200	infinite	0
10	200	200	200	10

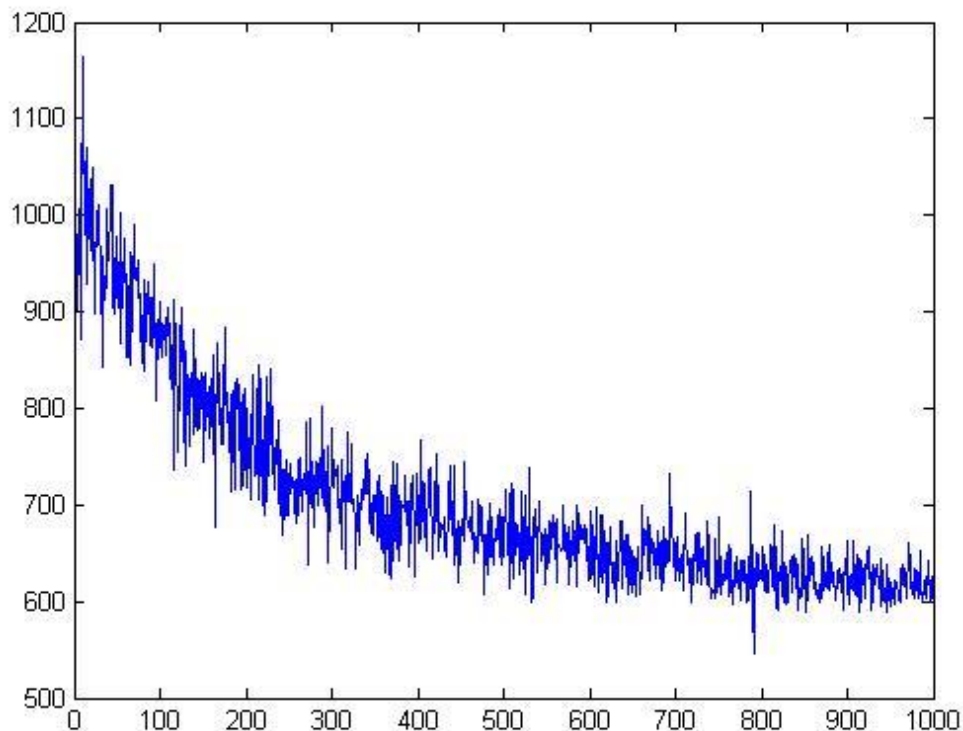
Δεδομένα προβλημάτων par.

PAR	Ants	N	p	q	a	b	Initial_C	Best_Sol	Opt	Pos
1	40	1000	0.1	1	1	2	710.33	545.22	524.61	792
6	40	1000	0.1	1	1	2	784.50	569.18	555.43	971
2	42	3000	0.1	0.5	1	2	1000.78	881.87	835.26	2808
7	42	3000	0.1	0.5	1	2	1141.73	970.08	909.68	459
3	45	1500	0.08	0.25	1	2	1096.10	849.22	826.14	397
8	45	1500	1	0.25	1	2	1242.49	896.10	865.94	498
12	45	1500	0.08	0.25	1	2	1146.44	856.76	819.56	258
14	45	1500	1	0.25	1	2	1197.61	893.01	866.37	1224
11	60	1650	0.05	0.75	1	2	1313.24	1078.30	1042.11	144
13	60	1650	0.07	0.75	1	2	1904.11	1621.11	1541.14	487
4	85	1800	0.05	0.5	1	2	1353.30	1084.33	1028.42	673
9	85	1800	0.05	0.5	1	2	1471.01	1200.55	1162.55	333
5	100	2000	0.01	1	1	2	1614.30	1327.63	1291.29	876
10	100	2000	0.01	1	1	2	1756.81	1415.26	1395.85	1670

Αποτελέσματα Ant Colony.

Σε γενικές γραμμές η ύπαρξη περιορισμών χρόνου, σε αντίθεση με αυτό που θα πιστεύαμε αρχικά, δεν προκαλεί πρόβλημα ούτε απαιτεί περισσότερο χρόνο για εύρεση ενός αποδεκτού αποτελέσματος, αυξάνει όμως πιθανότητες στασιμότητας σε τοπικό ελάχιστο. Επίσης παρατηρήθηκε σε μεγάλο αριθμό επαναλήψεων, ο καλύτερος συνδυασμός  $\alpha$   $\beta$  είναι 1 και 2 αντίστοιχα.

Ενδεικτικά κάποια παραδείγματα παρακάτω μας δείχνουν την πορεία προς το βέλτιστο.

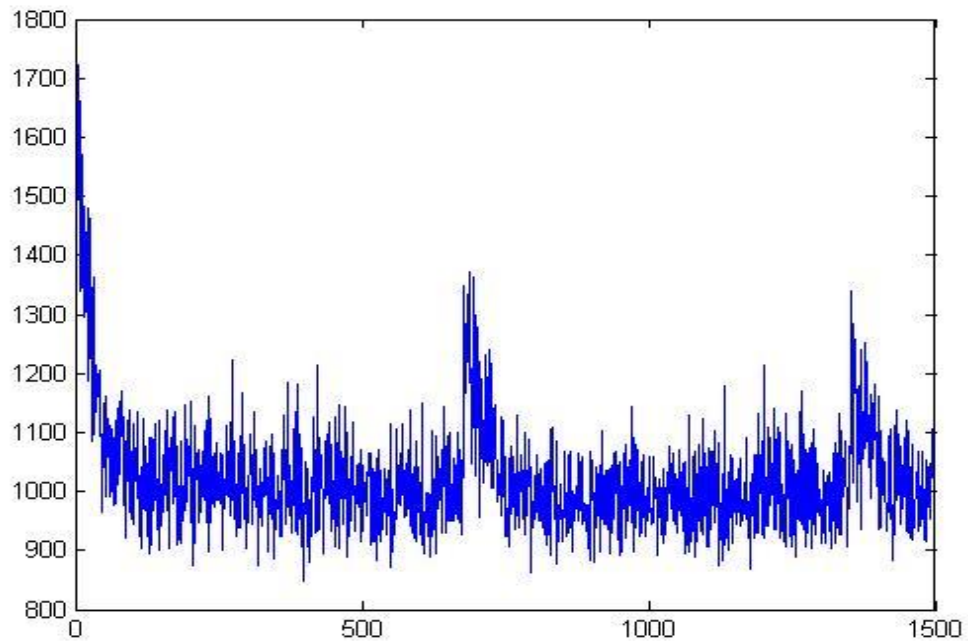


Διαδρομή παραδείγματος 1 προς βέλτιστο.

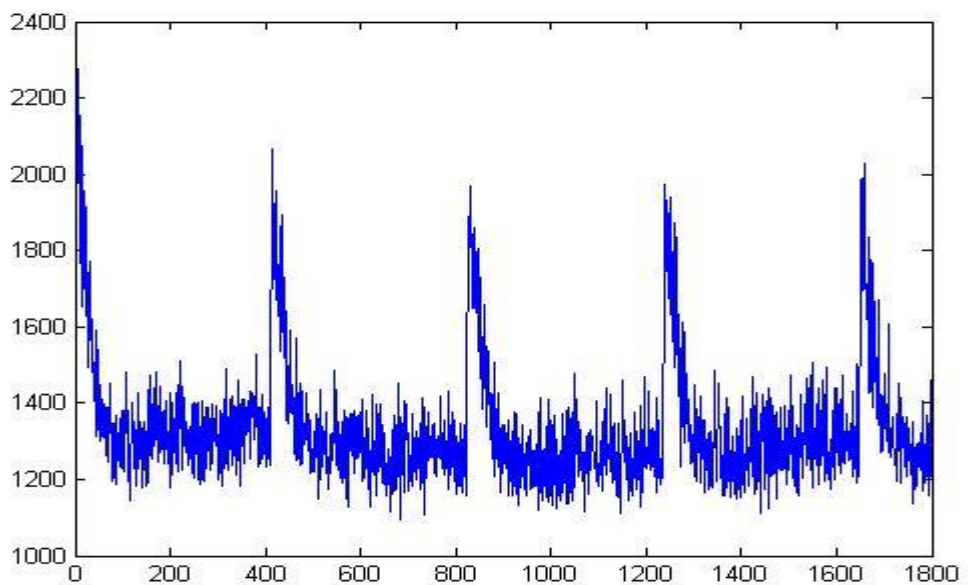
Το παραπάνω με ίχνος φερομόνης  $q=1$

Η μείωση του ίχνους φερομόνης σίγουρα θα απαιτήσει περισσότερες επαναλήψεις καθώς όπως θα παρατηρήσουμε στα 2 παρακάτω σχήματα των παραδειγμάτων 3 και 4 με  $q=0,25$  και  $q=0,5$  αντίστοιχα. Ο

λόγος είναι ότι σε τακτά χρονικά διαστήματα , οι πιθανότητες να επιλεγεί “κακό” μυρμήγκι και να χαλάσει προσωρινά το κόστος μας αυξάνονται με αποτέλεσμα να απαιτούνται περαιτέρω επαναλήψεις για επαναφορά.

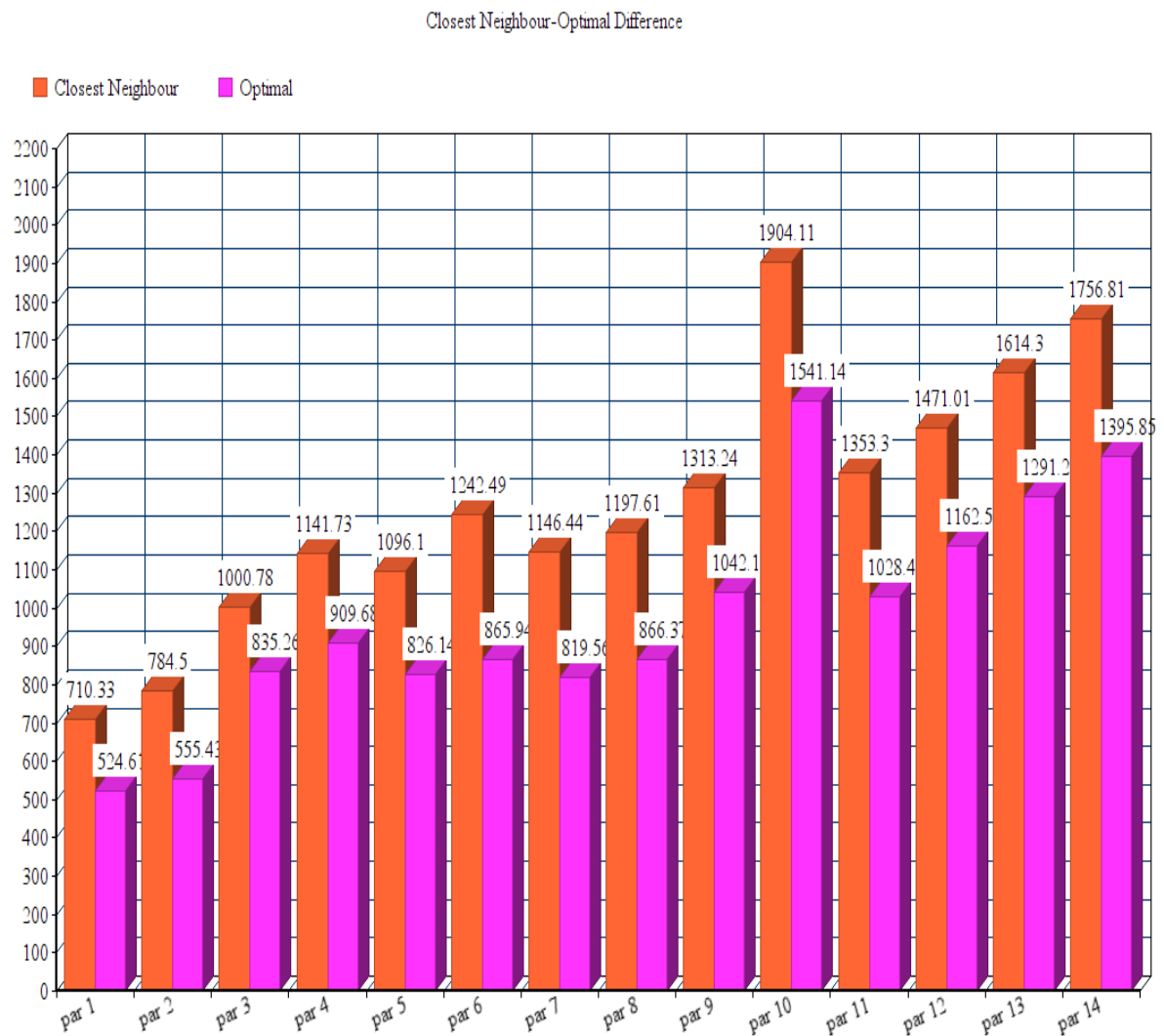


Διαδρομή παραδείγματος 3 προς βέλτιστο.



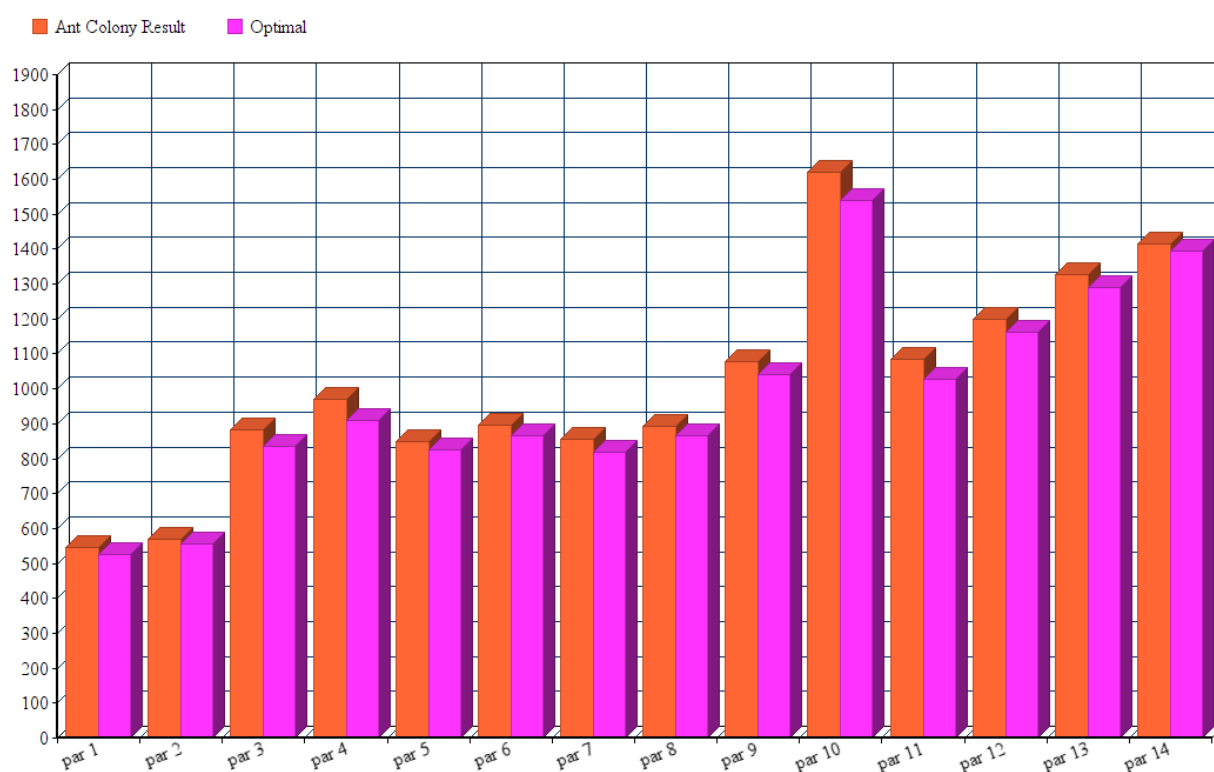
Διαδρομή παραδείγματος 4 προς βέλτιστο.

Η εφαρμογή του αλγορίθμου πλησιέστερου γείτονα έδωσε αποτελέσματα σχετικά καλά , σε απόσταση βέβαια από τα ποσοστά που θεωρούνται ικανοποιητικά γύρω από το βέλτιστο, όπως παρουσιάζονται και παρακάτω.



Διαφορά αποτελεσμάτων πλησιέστερου γείτονα σε σχέση με το γνωστό βέλτιστο.

Closest Neighbour-Optimal Difference



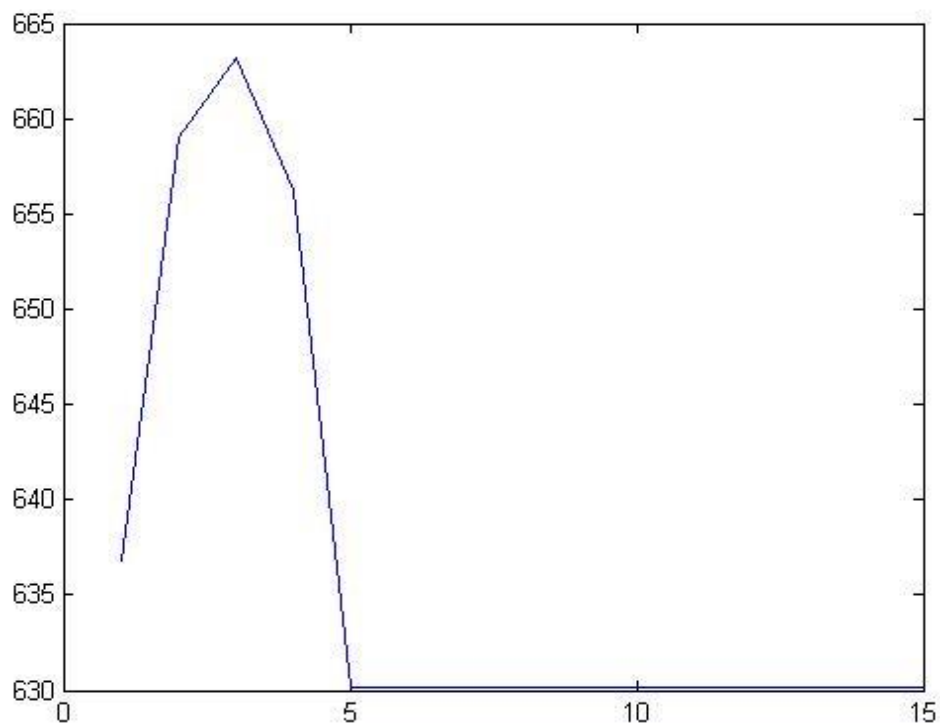
Διαφορά AntColony σε σχέση με το γνωστό βέλτιστο.

PAR	Diff from Optimal with Ant Colony	Diff from Optimal with Closest Neighbor
1	3.85%	30.02%
6	2.44%	34.19%
2	5.36%	18.03%
7	6.42%	22.62%
3	2.75%	28.08%
8	3.42%	35.70%
12	4.43%	33.25%
14	3.02%	32.04%
11	3.41%	23.02%
13	5.05%	21.06%
4	5.29%	27.28%
9	3.21%	23.42%
5	2.73%	22.24%
10	1.38%	22.89%

Ποσοστιαία διαφορά βέλτιστου από Ant Colony και Closest Neighbor Algorithm.

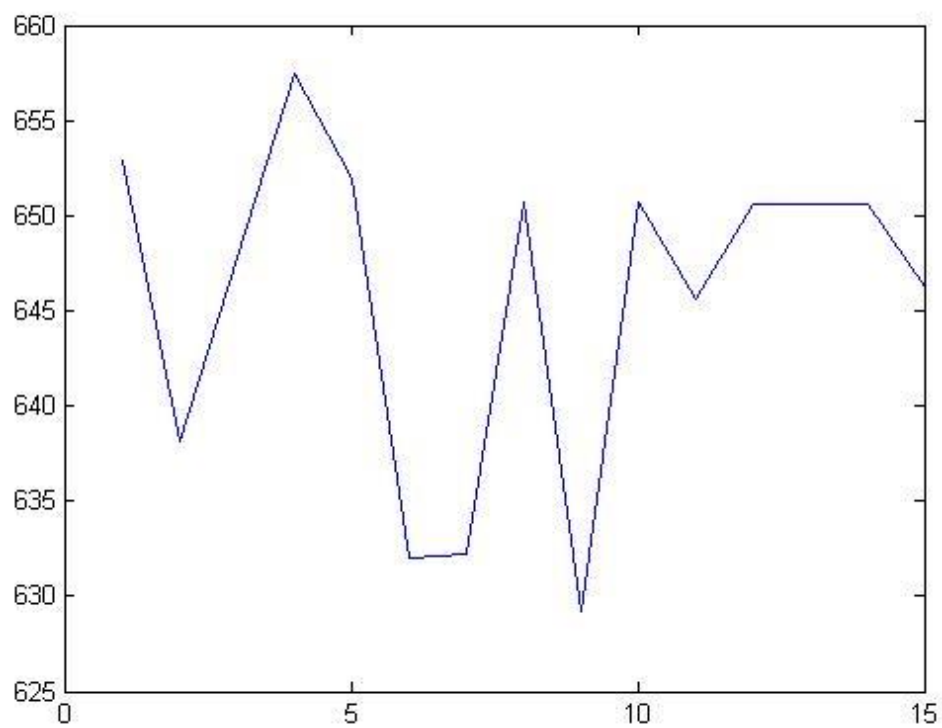
Σε γενικές γραμμές η διαφορά είναι πολύ μικρή. Απλή μετατόπιση ενός κόμβου ενδέχεται να κλείσει την διαφορά σε μηδαμινό ή μηδενικό επίπεδο. Σε κάθε περίπτωση, χρόνου επιτρέποντος, δεν αποκλείει κανείς ότι με πολύ χρονοβόρες και πολύ περισσότερες συνεπώς επαναλήψεις, μπορεί να βρεθεί το βέλτιστο ακριβώς.

Παρακάτω παρουσιάζεται η πορεία που ακολουθεί το παράδειγμα 1, σε περίπτωση εντελώς αντίθετη από αυτήν που ακολουθήσαμε. Ο λόγος για την γρήγορη αλλά μη αποτελεσματική σύγκλιση με μεγάλες τιμές  $a$  και  $b$  με σταθερά  $p=0,7$  και  $q=0,5$ , όλα για 15 επαναλήψεις και 25 Ants.

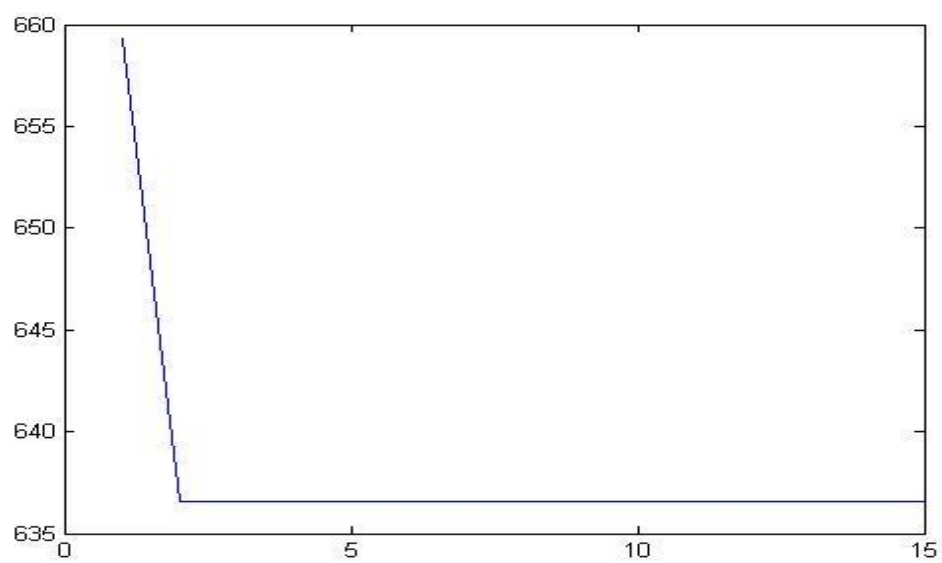


$a=5$  ,  $b=6$ .



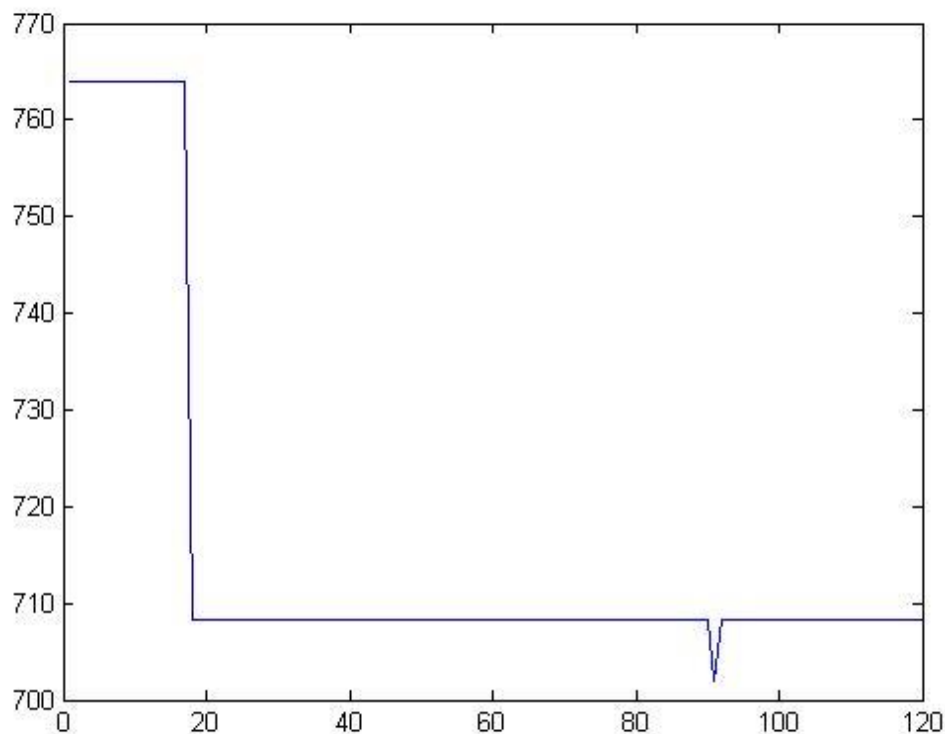


**a=10 ,b=12.**



**a=25 ,b=27.**

Η σημαντικότερη παρατήρηση εδώ, είναι ότι από την πρώτη κιόλας επανάληψη φτάνουμε σε τιμές που με το  $\alpha=1$  και  $\beta=2$  που χρησιμοποιήσαμε χρειαστήκαμε πάνω από 200 επαναλήψεις. Παρόλα αυτά η τιμή αυτή δεν μειώνεται και κολλάει όπως βλέπουμε στην παραπάνω εικόνα σε τοπικά ελάχιστα. Οι τεράστιες τιμές ( $>25$ ) για τις παραμέτρους  $\alpha$ ,  $\beta$  μας δίνουν σταθερό αποτέλεσμα σε όλες τις επαναλήψεις σχεδόν.



**$a=b=35$   $\rho=0,1$   $q=0.5$  για 120 επαναλήψεις 25Ants.**

Είναι φανερό ότι κολλάει σε τιμή που βρίσκεται κοντά στο αποτέλεσμα του πλησιέστερου γείτονα και μένει εκεί, εκτός ελαχίστων, μη ικανοποιητικών εξαιρέσεων.

Στην συνέχεια ακολουθήθηκε μια διαφορετική τακτική με αυξημένα  $\alpha$  και  $\beta$  για γρηγορότερη σύγκλιση καθώς το πλήθος των κόμβων είναι πολύ μεγάλο και η αργή σύγκλιση με 1000 και πλέον επαναλήψεις δεν είναι πρακτική λύση.

Kel	Nodes	Capacity	Route_Time	Service_Time
1	240	550	650	0
2	320	700	900	0
3	400	900	1200	0
4	480	1000	1600	0
5	200	900	1800	0
6	280	900	1500	0
7	360	900	1300	0
8	440	900	1200	0
9	255	1000	infinite	0
10	323	1000	infinite	0
11	399	1000	infinite	0
12	483	1000	infinite	0
13	252	1000	infinite	0
14	320	1000	infinite	0
15	396	1000	infinite	0
16	480	1000	infinite	0
17	240	200	infinite	0
18	300	200	infinite	0
19	360	200	infinite	0
20	420	200	infinite	0

Δεδομένα προβλημάτων kel.

Kel	Ants	N	p	q	a	b	Initial_C	Best_Sol	Opt	Pos
1	120	25	0.50	1	5	6	7703.5	6243.5	5627.54	12
2	160	25	0.50	1	4	4	12030	9752.7	8444.50	3
3	200	20	0.40	1	2	3	15589	12445.3	11036.2	8
4	240	20	0.40	1	2	3	18101	15138.7	13624.5	8
5	100	25	0.50	1	5	6	7604	6532.6	6460.98	2
6	140	30	0.55	1	3	4	10864	9932.4	8412.80	19
7	180	25	0.50	1	1	2	12864	10788.2	10181.7	1
8	220	20	0.40	1	3	3	15887	12348.6	11643.9	5
9	127	25	0.50	1	5	6	716.53	617.9	583.39	2
10	161	30	0.55	1	5	6	905.33	821.6	741.56	9
11	199	30	0.55	1	5	6	1125.1	1000.7	918.45	12
12	241	20	0.40	1	5	6	1392.04	1233.1	1107.19	2
13	126	25	0.50	1	5	6	992.75	949.3	859.11	6
14	160	20	0.40	1	4	4	1308.6	1153.8	1081.31	15
15	198	25	0.50	1	5	8	1566.7	1422.6	1345.23	9
16	240	23	0.45	1	5	5	1921.9	1711.1	1622.69	4
17	120	30	0.55	1	5	6	884.72	767.52	707.79	12
18	150	30	0.6	1	5	6	1283.3	1078.45	997.52	8
19	180	18	0.7	1	4	4	1742.6	1432.7	1366.86	4
20	210	18	0.7	1	5	6	2204.3	2018.5	1820.09	4

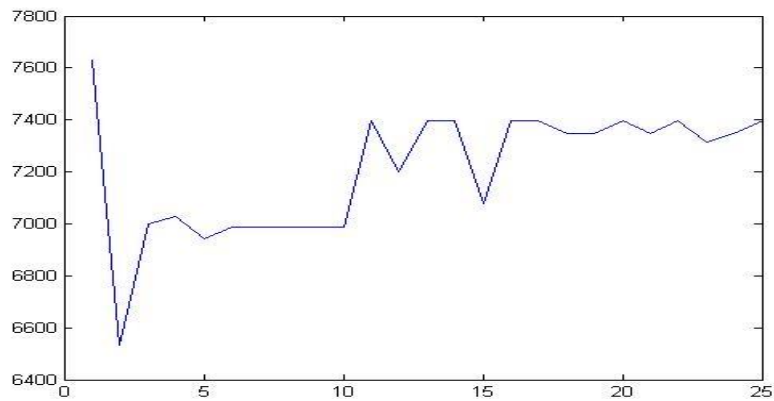
Αποτελέσματα Ant Colony.

Στα παραπάνω παραδείγματα και σε αντίθεση με τα προηγούμενα 14, χρησιμοποιήθηκαν λιγότερες επαναλήψεις με κάπως μεγαλύτερα  $\alpha, \beta$  για πιο γρήγορη σύγκλιση. Αυτό έγινε σε συνδυασμό με αρκετά μυρμήγκια πλήθους 50% των κόμβων.

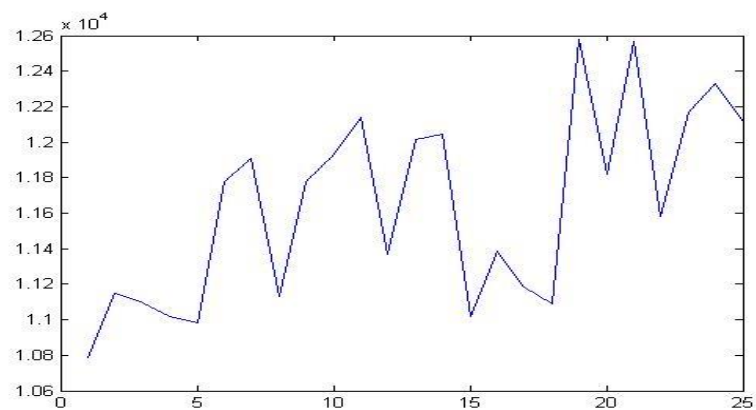
Στο προηγούμενο Set παραδειγμάτων δεν φάνηκε περισσότερα από τα μισά μυρμήγκια να βοηθάν το αποτέλεσμα ιδιαίτερα.

Σχετικά με τον χρόνο δεν βοήθησε φυσικά γιατί ναι μεν οι επαναλήψεις είναι λίγες αλλά ο παράγοντας **Ptable** και το πλήθος των υπολογισμών που απαιτούνται όσο αυξάνονται οι κόμβοι το  $\alpha$  και το  $\beta$ , αυξάνουν αντιστοίχως και τον χρόνο για εύρεση αποτελέσματος.

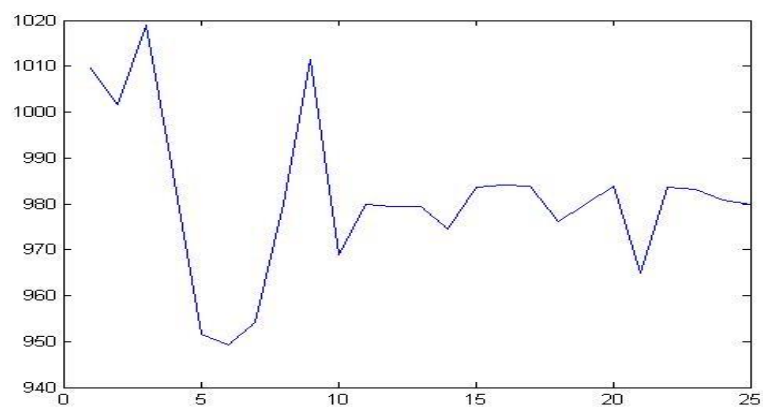
Σε συνέχεια των προηγούμενων παραδειγμάτων, παρουσιάζονται ενδεικτικά οι πορείες κάποιων *kel* παραδειγμάτων προς το βέλτιστο αλλά και η ποσοστιαία διαφορά του από τις μεθόδους του ClosestNeighbor και του AntColony.



Kel05.



Kel07.

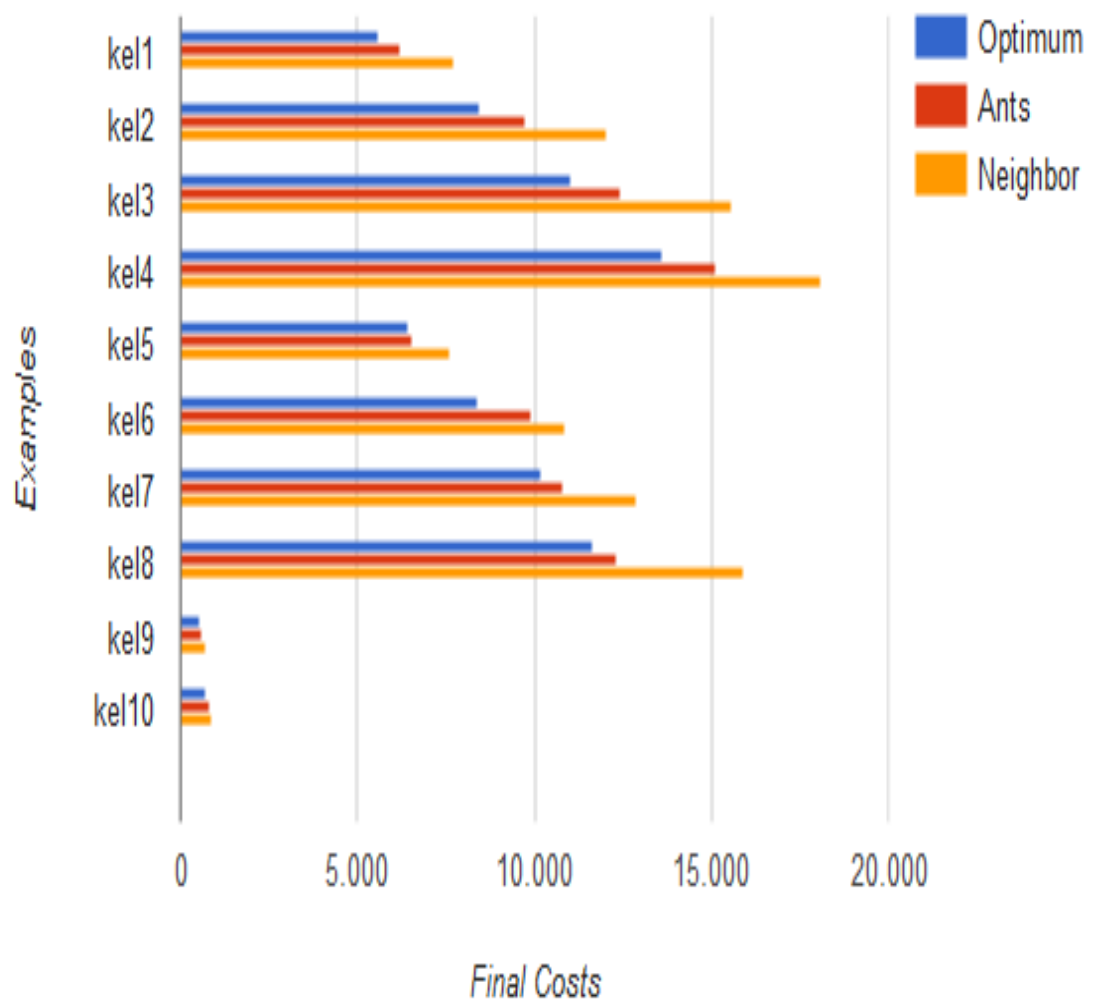


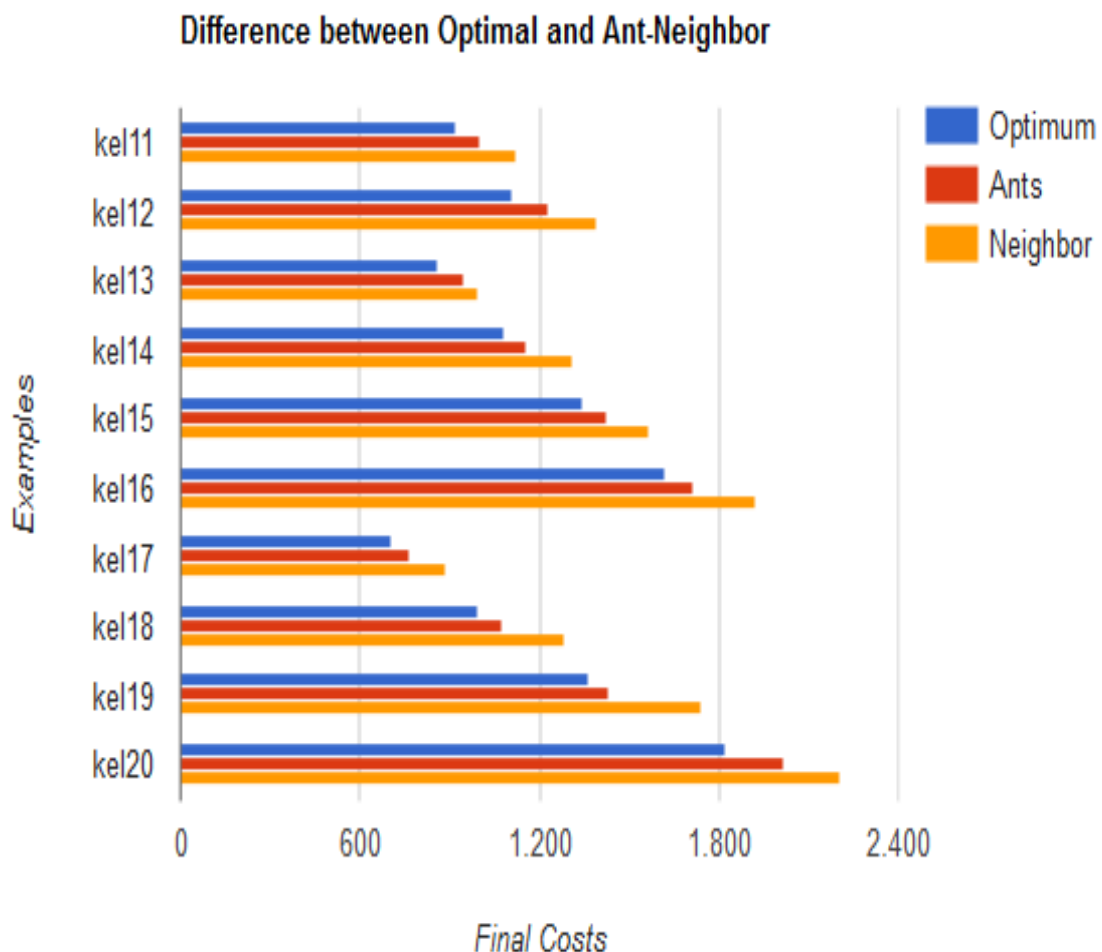
Kel13.

KEL	Diff from Optimal with Ant Colony	Diff from Optimal with Closest Neighbor
1	10.37%	31.14%
2	14.37%	35.02%
3	12.08%	34.19%
4	10.52%	28.22%
5	1.10%	16.25%
6	16.51%	25.43%
7	5.78%	23.27%
8	5.87%	30.82%
9	5.74%	20.48%
10	10.24%	19.88%
11	8.57%	20.22%
12	10.76%	22.79%
13	9.97%	14.43%
14	6.48%	19.02%
15	5.59%	15.21%
16	5.30%	16.88%
17	8.09%	22.22%
18	7.79%	25.05%
19	4.70%	24.16%
20	10.33%	19.09%

Ποσοστιαία διαφορά βέλτιστου από Ant Colony και Closest Neighbor Algorithm.

Difference between Optimal and Ant-Neighbor





Εν κατακλείδι αυτό που μπορούμε να πούμε είναι:

- Όσο μεγαλώνει το μήκος της διαδρομής, τόσο περισσότερο απέχουμε από την βέλτιστη λύση με σχετικά λίγες επαναλήψεις.
- Με μεγαλύτερες τιμές των  $\alpha$   $\beta$  παρατηρήσαμε αποκλίσεις ανάμεσα στις λύσεις, ανεξαρτήτως πλήθους κόμβων.
- Ο χρόνος επίλυσης για κάθε παράδειγμα είναι πολύ εύκολο να φτάσει σε μεγάλα επίπεδα(>50 min) χωρίς πάντα να αποτελεί αυτό εγγύηση βέλτιστης λύσης.

Οι παράμετροι  $\alpha$   $\beta$   $\rho$  και  $q$  είναι ξεχωριστοί για κάθε παράδειγμα. Μπορεί εδώ να μην εξατομικεύτηκαν πλήρως, πάντως το κάθε παράδειγμα χρειάζεται έναν ξεχωριστό συνδυασμό των 4 αυτών, συνεπώς τίθεται κάπως ένα θέμα βελτιστοποίησης και σε αυτό το κομμάτι.



## Βιβλιογραφία

1. A Green Vehicle Routing Problem, Sevgi Erdogan, Else Miller-Hooks, 2011.
2. A Column Generation Approach for a Multi-Attribute Vehicle Routing Problem, Imam Dayarian, Teodor Gabriel Grainic, Michel Gendreau, Walter Rei, 2013.
3. A Generic Heuristic for Vehicle Routing Problems with Multiple Synchronization Constraints, Michael Drexler, 2014.
4. Strategic Analysis of the Dairy Transportation Problem, Nadia Lahrichi, Teodor Gabriel Grainic, Michel Gendreau, Walter Rei, Louis-Martin Rousseau, 2012.
5. The Relationship between Vehicle Routing & Scheduling and Green Logistics –A Literature Survey, A. Sbihi, R.W. Eglese, 2007.
6. Μεθοδολογίες Ανάπτυξης Πληροφοριακών Συστημάτων Υποστήριξης Λήψης Αποφάσεων για την Δρομολόγηση Στόλου Οχημάτων και την Αξιολόγηση Εντολών Μεταφοράς Φορτίου στον Τομέα των Logistics. Διδακτορική Διατριβή, Γεώργιος Ν. Μουρκούσης, 2008.
7. Σχεδιασμός ενός Πληροφοριακού Συστήματος για τη Δρομολόγηση Στόλου Οχημάτων με Εφαρμογή σε Μεταφορική Εταιρεία, Βασίλειος Πουπούζας, 2007.
8. Abdelazziz, M.M., El-Ghareeb, H.A., Ksasy, M.S.M. (2014), Hybrid heuristic algorithm for solving capacitated vehicle routing problem, International journal of computers and technology, Vol. 12 No.9, pp. 3844-3851.
9. Ali, N., Othman, M.A., Husain, M.N., Misran, M.H. (2014), A review of firefly algorithm, ARPN journal of engineering and applied sciences, Vol. 9 No. 10, pp. 1732-1736 .

10. Abshouri, A., Meybodi, M., Bakhtiary, A. (2011), New firefly algorithm based on multi swarm & learning automata in dynamic environments, in: IEEE proceedings, pp. 73-77.
11. Baldacci, R., Battarra, M., Vigo, D. (2008), Routing a heterogeneous fleet of vehicles, in: Golden, B., Raghavan, S., Wasil, E., (Eds.), The vehicle routing problem: Latest advances and new challenges, Springer, pp. 3-27 .
12. Brandao, J., (2011), A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, Computers & Operations Research, Vol. 38 No.1, pp. 140-151 .
13. S. Ruland and E.Y. Robin. The pickup delivery problem: Faces and Branch-and-Cut algorithm. Computers Mathematical Applications, 33 (12): 1- 13, 1997.
14. R. Thangiah, J.Y. Potvin, and T. Sung. Heuristics approaches to vehicle routing with backhauls and time windows. Computers and Operations Research, 23: 1043- 1057, 1996.
15. Ιωάννης Μαρινάκης, Μαγδαληνή Μαρινάκη, σημειώσεις μεταπτυχιακού μαθήματος: Εξελικτικοί Αλγόριθμοι και Βελτιστοποίηση Συστημάτων μεγάλης κλίμακας. **Φεβρουάριος 2010.**
16. Deneubourg, J.-L., S. Aron, S. Goss and J.M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant", Journal of Insect Behavior 3, 1990, 159-168 .
17. Eric D. Taillard (1999), Ant systems, Technical Report IDSIA-05-99.
18. Μάρκος Παπαγεωργίου, σημειώσεις μαθήματος : Μη γραμμικός προγραμματισμός, 2015.

