# ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

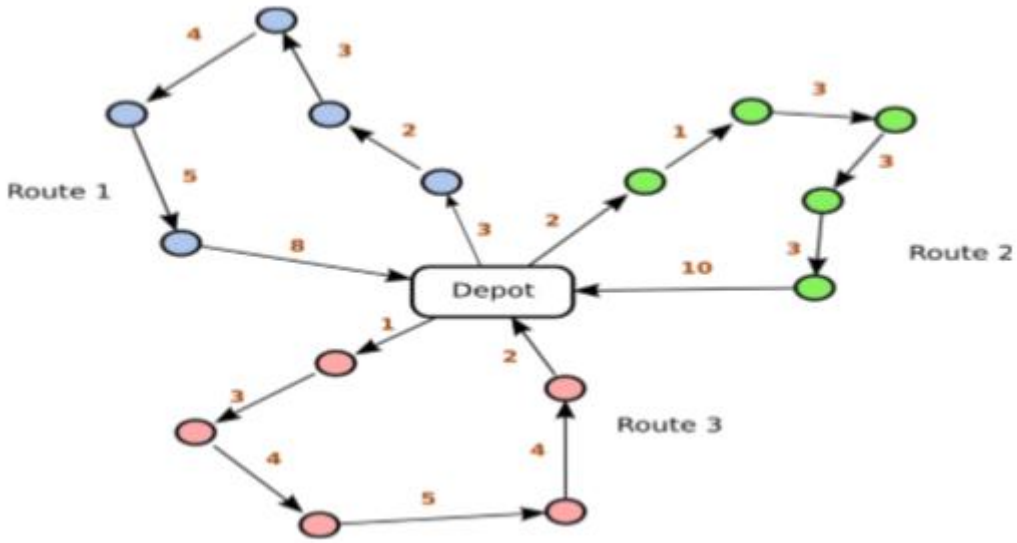
## ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ


**Εργασία Μεταπτυχιακού Διπλώματος**

**Επίλυση του Σωρευτικού Προβλήματος Δρομολόγησης Οχημάτων με αυτοπροσδιοριζόμενο αλγόριθμο της αποικίας μυρμηγκιών**


**Κυριακάκης Νικόλαος-Αντώνιος**

Επιβλέπων Καθηγητής
**Ιωάννης Μαρινάκης**

## *Ευχαριστίες*

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Μαρινάκη Ιωάννη, για όλη τη βοήθεια του, την καθοδήγηση και τις επισημάνσεις του πάνω στην παρούσα μεταπτυχιακή εργασία, αλλά κυρίως διότι αποτέλεσε την αιτία που επέλεξα να ασχοληθώ με το συγκεκριμένο επιστημονικό πεδίο και να συνεχίσω τις μεταπτυχιακές μου σπουδές στη Σχολή Μηχανικών Παραγωγής και Διοίκησης.

Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές μου, που μέσω των μαθημάτων τους μου άνοιξαν νέους ορίζοντες, διευρύνοντας τα σύνορα των γνώσεων μου, δίνοντας μου ουσιαστικά μια νέα οπτική των πραγμάτων.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη στήριξη τους, την εμπιστοσύνη τους και το κουράγιο τους. Τους φίλους μου για τις ωραίες στιγμές των φοιτητικών μας χρόνων και τις εποικοδομητικές συζητήσεις μας για το μέλλον μας στον εργασιακό στίβο.

# Περιεχόμενα

# ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

## 1.1 Η εφοδιαστική αλυσίδα

Στον σύγχρονο κόσμο όπου πρώτες ύλες, ενδιάμεσα και τελικά προϊόντα διακινούνται από και προς κάθε γωνιά του κόσμου, η επιστήμη της επιχειρησιακής έρευνας και πιο συγκεκριμένα το κομμάτι που ασχολείται με την εφοδιαστική αλυσίδα τείνει να παίξει καθοριστικό ρόλο στη δημιουργία πλεονεκτήματος έναντι του ανταγωνισμού. Η Διαχείριση εφοδιαστικής αλυσίδας ορίζεται ως η διαχείριση ενός δικτύου εσωτερικά συνδεδεμένων επιχειρήσεων που συμμετέχουν στην απώτερη παροχή πακέτων προϊόντων και υπηρεσιών, τα οποία απευθύνονται στους τελικούς καταναλωτές (Harland, 1996). Η διαχείριση εφοδιαστικής αλυσίδας εκτείνεται σε όλη τη διαδικασία μεταφοράς και αποθήκευσης των πρώτων υλών, απογραφής της εσωτερικής διαδικασίας και παροχής ολοκληρωμένων αγαθών από πλευράς προέλευσης μέχρι και την κατανάλωση τους. Περιλαμβάνει όλα τα ενδιάμεσα στάδια που στόχο έχουν την ικανοποίηση των απαιτήσεων των πελατών. Αποτελείται από προμηθευτές, κατασκευαστές, χώρους αποθήκευσης, αποθέματα, κέντρα διανομής, μεταφορείς, πωλητές, έτοιμα προϊόντα και πελάτες.

Τα ενδιάμεσα στάδια της διανομής και συγκεκριμένα οι αποφάσεις που πρέπει να ληφθούν για αυτά, δημιουργούν επιμέρους προβλήματα που η επιστήμη της εφοδιαστική αλυσίδας καλείται να επιλύσει κατά το βέλτιστο δυνατό τρόπο. Τέτοια προβλήματα αποτελούν, η διαμόρφωση του δικτύου διανομής, δηλαδή ο αριθμός, θέση και το δίκτυο αποστολών των προμηθευτών, των εγκαταστάσεων παραγωγής, των κέντρων διανομής, των αποθηκών, των αποβάθρων και των πελατών. Η στρατηγική διανομής, όπου λαμβάνονται αποφάσεις σχετικά με τον τρόπο ελέγχου των λειτουργιών, τα συστήματα διανομής και τα μέσα. Η αρμονία μεταξύ των επιμέρους δραστηριοτήτων της αλυσίδας και η συμβατότητα μεταξύ αυτών, με στόχο τη συνολική βελτιστοποίηση της. Η διακίνηση των πληροφοριών μέσα στην αλυσίδα, με στοιχεία για την ζήτηση, τις προβλέψεις, το απόθεμα, τη μεταφορά . Η διαχείριση αποθεμάτων, που αφορούν την ποσότητα και τον τόπο πρώτων υλών, ενδιάμεσων και τελικών προϊόντων. Οι χρηματικές ροές, δηλαδή οι αποφάσεις που αφορούν τις πληρωμές και την διακίνηση κεφαλαίων μεταξύ των μερών της αλυσίδας.

Η διαμόρφωση του δικτύου διανομής αποτελείται από επιμέρους υποπροβλήματα, τα οποία χωρίζονται σε τρεις μεγάλες κατηγορίες. Τα προβλήματα δρομολόγησης, τα προβλήματα χρονοπρογραμματισμού και τα προβλήματα χωροθέτησης. Κάθε κατηγορία είναι και μια ομάδα προβλημάτων στα οποία υπάρχουν παραλλαγές. Για παράδειγμα, η ύπαρξη χρονικών παραθύρων, η ζήτηση σε πραγματικό χρόνο, η ταυτόχρονη διανομή και τα πολλαπλά προϊόντα. Σε κάθε εφοδιαστική αλυσίδα ενδέχεται να συνυπάρχουν ταυτόχρονα και τα τρία προβλήματα, των οποίων οι λύσεις θα πρέπει να συνδυάζονται κατάλληλα, ώστε να βελτιστοποιούν το δίκτυο διανομής στο σύνολό του.

Η εργασία αυτή αφορά την πρώτη κατηγορία προβλημάτων, αυτή της δρομολόγησης οχημάτων. Σε αυτό το είδος προβλημάτων αναζητείται η διαδρομή που θα πρέπει να ακολουθήσει κάθε όχημα, ώστε να ελαχιστοποιηθεί το συνολικό κόστος των διαδρομών όλων των οχημάτων. Η δρομολόγηση υπόκειται ταυτόχρονα σε περιορισμούς, με πιο συχνό αυτόν της ζήτησης, η οποία θα πρέπει να ικανοποιείται και αυτός της χωρητικότητας του οχήματος. Όπως προαναφέρθηκε, οι παραλλαγές είναι πολλές και αφορούν κυρίως στους περιορισμούς που θα πρέπει να ληφθούν υπόψη. Η εργασία μελετά και επιλύει μια τέτοια παραλλαγή, το Συσσωρευτικό Πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας, το οποίο παρουσιάζεται αναλυτικά στο επόμενο κεφάλαιο.

# ΚΕΦΑΛΑΙΟ 2: ΣΥΣΣΩΡΕΥΤΙΚΟ ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (CCVRP)

## 2.1 Εισαγωγή

Στην μεγάλη οικογένεια των προβλημάτων δρομολόγησης δυο είναι τα πιο διαδεδομένα και ευρέως γνωστά. Το πρόβλημα του πλανόδιου πωλητή (TSP) και το απλό πρόβλημα δρομολόγησης οχημάτων(CVRP). Στο πρόβλημα του πλανόδιου πωλητή, το ζητούμε είναι η εύρεση της συντομότερης διαδρομής όπου ο "πωλητής" περνάει από όλους τους κόμβους μία φορά ακριβώς και επιστρέφει στον αρχικό. Το πρόβλημα δρομολόγησης οχημάτων προτάθηκε από τους Dantzig and Ramser το 1959 και είναι η βάση για τα περισσότερα προβλήματα δρομολόγησης. Περιλαμβάνει την εξυπηρέτηση πελατών με ένα στόλο από οχήματα. Στόχος είναι η ελαχιστοποίηση του συνολικού κόστους διαδρομής.

Έχοντας ως κορμό το απλό CVRP στη βιβλιογραφία έχουν διατυπωθεί πολλά προβλήματα δρομολόγησης. Γνωστότερα είναι: το πρόβλημα με παραλαβή και παράδοση (VRPPD), το πρόβλημα με χρονικά παράθυρα (VRPTW), το πρόβλημα με πολλαπλές διαδρομές (VRPMT) και το ανοιχτό πρόβλημα δρομολόγησης (OVRP). Αυτά συναντώνται σε πάρα πολλές καθημερινές διαδικασίες των επιχειρήσεων και η επίλυση τους μπορεί να καθορίσει την βιωσιμότητα τους.

Σε πολλές περιπτώσεις στην καθημερινότητα τα προβλήματα που καλούμαστε να επιλύσουμε ξεπερνούν τα κλασικά προβλήματα ελαχιστοποίησης κόστους καθώς σε αντίθεση με αυτά που έχουν σαν στόχο την μεγιστοποίηση του κέρδους παραβλέπουν την ανάγκη για γρήγορη εξυπηρέτηση και παροχή κρίσιμων αγαθών μετά από κάποια καταστροφή και σε περιπτώσει έκτακτης ανάγκης. Ενώ οι εμπορικές εφοδιαστικές αλυσίδες εστιάζουν στην ποιότητα και την κερδοφορία, οι ανθρωπιστικές εφοδιαστικές αλυσίδες έχουν ως πρώτο στόχο την ελαχιστοποίηση των ανθρώπινων απωλειών. Για το λόγο αυτό είναι πιο ανθρωποκεντρικές. Σε αυτά τα προβλήματα το κριτήριο που υπερισχύει στην αξιολόγηση μιας λύσης είναι ο χρόνος στον οποίο θα εξυπηρετηθεί ο κάθε πελάτης. Δηλαδή, λαμβάνεται υπόψη στη

κοστολόγηση της λύσης η σειρά με την οποία ο συγκεκριμένος πελάτης εξυπηρετήθηκε. Το πρόβλημα που μοντελοποιεί το συγκεκριμένο ζητούμενο είναι το Συσσωρευτικό Πρόβλημα δρομολόγησης που περιγράφεται αναλυτικά σε αυτό το κεφάλαιο.

Το Συσσωρευτικό Πρόβλημα δρομολόγησης οχημάτων είναι ένα NP-Hard πρόβλημα που έχει ως στόχο την ελαχιστοποίηση του χρόνου άφιξης στους πελάτες, αντί για την κλασική ελαχιστοποίηση της απόστασης, έχοντας ως περιορισμό την χωρητικότητα του οχήματος. Το CCVRP είναι ουσιαστικά μια παραλλαγή του Traveling Repairman Problem, με την προσθήκη του περιορισμού χωρητικότητας και την ύπαρξης ομοιογενούς στόλου οχημάτων.

## 2.2 Μοντελοποίηση

Θεωρούμε το πρόβλημα CCVRP ορισμένο σε ένα γράφημα $G=(V,E,W)$ όπου $V=\{0,1,2,\ldots,n+1\}$ οι κόμβοι του γραφήματος (οι κόμβοι 0,n+1 είναι η αποθήκη), $V'=V/\{0,n+1\}$ οι κόμβοι των πελατών, Ε είναι το διάνυσμα των τόξων σε κάθε ένα εκ των οποίων αντιστοιχεί ένα βάρος $w_{ij}=w_{ji}\in W$ που αντιστοιχεί στο κόστος ή την χρονική απόσταση ανάμεσα στους κόμβους $i$ και $j$. Άλλοι παράμετροι του προβλήματος είναι :

- $\boldsymbol{q_i}$, η ζήτηση του κόμβου $i$
- $\boldsymbol{m}$, ο αριθμός των όμοιων οχημάτων
- $\boldsymbol{Q}$, η χωρητικότητα των οχημάτων

Το αντικείμενο του CCVRP είναι η εύρεση των δρομολογίων κατά τέτοιο τρόπο ώστε κάθε πελάτης να εξυπηρετείται μόνο μια φορά και το άθροισμα των χρόνων άφιξης σε αυτούς να ελαχιστοποιείται. Ένα δρομολόγιο ορίζεται ως ένα κύκλωμα που αρχίζει και τελειώνει στην αποθήκη, με τη συνολική ζήτηση των επισκεπτόμενων κόμβων να μην ξεπερνά την χωρητικότητα $Q$. Ορίζουμε $t_i^{\kappa}$ το χρόνο άφιξης του οχήματος $k$ στον κόμβο $i$ και $x_{ij}$ τη δυαδική μεταβλητή ίση με 1 αν το όχημα μεταβαίνει από τον κόμβο $i$ στον $j$. Παρακάτω γίνεται η μαθηματική απεικόνιση του προβλήματος και παρουσιάζονται οι περιορισμοί του.

## 2.3 Μαθηματική αναπαράσταση

**Αντικειμενική συνάρτηση κόστους :**

$$F : Min \sum_{k=1}^{m} \sum_{i \in V'} t_i^k \quad (1)$$

**Περιορισμοί :**

$$\sum_{i \in V} x_{ij}^{k} = \sum_{i \in V} x_{ji}^{k}, \forall j \in V', \forall j \in [1....\mathrm{m}] (2)$$

$$\sum_{k=1}^{m} \sum_{j \in V} x_{ij}^{k} = 1, \forall i \in V' (3)$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^{k} q_i \leq Q, \forall k \in [1,...,\mathrm{m}] (4)$$

$$\sum_{j \in V} x_{0j}^{k} = 1, \forall k \in [1,...,\mathrm{m}] (5)$$

$$\sum_{i \in V} x_{i,n+1}^{k} = 1, \forall k \in [1,...,\mathrm{m}] (6)$$

$$t_i^k + w_{ij} - (1 - \mathrm{x}_{ij}^{k})\mathrm{T} \leq \mathrm{t}_j^k, \forall i \in V / \{\mathrm{n+1}\}, \forall j \in V, \forall k \in [1,.,\mathrm{m}] (7)$$

$$t_i \geq 0, \forall i \in V, \forall k \in [1,...,\mathrm{m}] (8)$$

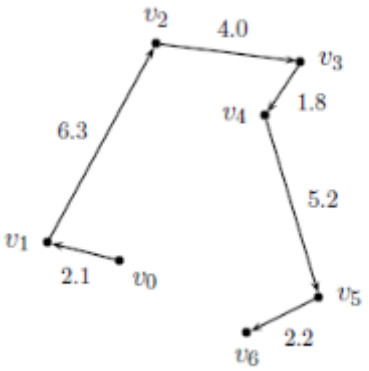
$$x_{ij} = \{0,1\}, \forall i \in V, i \neq j, \forall k \in [1,...,m] (9)$$

$$x_{ij} = 0 \acute{\eta} 1, (\mathrm{i}, \mathrm{j}) \in \mathrm{A} (10)$$

Αναλυτικά, η αντικειμενική εξίσωση (1) αποτελεί την προς ελαχιστοποίηση συνάρτηση κόστους. Οι περιορισμοί (2) καθορίζουν ότι τα οχήματα που επισκέπτονται τον κόμβο *i* πρέπει να αποχωρήσουν από αυτόν. Οι περιορισμοί (3) εξασφαλίζουν ότι κάθε πελάτης εξυπηρετείται μία φορά ακριβώς. Οι περιορισμοί (4) είναι περιορισμοί χωρητικότητας. Ο συνολικός φόρτος της διαδρομής δεν μπορεί να υπερβαίνει την χωρητικότητα τους οχήματος. Οι περιορισμοί (5) και (6) εξασφαλίζουν ότι κάθε διαδρομή αρχίζει και τελειώνει στην αποθήκη. Τέλος οι (7) υπολογίζουν τον χρόνο άφιξης σε κάθε κόμβο και εξασφαλίζουν ότι δεν θα υπάρχουν υπό-διαδρομές με τη βοήθεια ενός μεγάλου αριθμού $T \gg t_i^k + w_{ij}$.

Για την καλύτερη κατανόηση του τρόπου με τον οποίο κοστολογείται μια λύση δίνεται το ακόλουθο παράδειγμα:

-Έστω λύση η διαδρομή $U = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_0]$.




-Ο τύπος υπολογισμού του κόστους είναι : $f(x) = \sum_{i=1}^{n} (\mathrm{n} - \mathrm{i} + 1)\, \mathrm{w}(\mathrm{x}_{i-1}, \mathrm{x}_i)$

-Υπολογισμός του κόστους :

$$f(x) = 6(2.1) + 5(6.3) + 4(4.0) + 3(1.8) + 2(5.2) + 1(2.2) = 78.1$$

# ΚΕΦΑΛΑΙΟ 3 : Ο ΑΛΓΟΡΙΘΜΟΣ ΤΗΣ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ (ACO)

## 3.1 Αλγόριθμοι

### 3.1.1 Αλγόριθμοι βελτιστοποίησης

Οι περισσότεροι συμβατικοί ή κλασικοί αλγόριθμοι βελτιστοποίησης είναι ντετερμινιστικοί ή αλλιώς αιτιοκρατικοί. Δηλαδή κάθε δράση που λαμβάνει χώρα συνδέεται με αιτιώδη αλυσιδωτή σχέση με τις προηγούμενες κατά απόλυτο τρόπο. Για παράδειγμα η μέθοδος simplex που χρησιμοποιείται στην επίλυση προβλημάτων γραμμικού προγραμματισμού είναι αιτιοκρατική. Μερικοί αιτιοκρατικοί αλγόριθμοι βελτιστοποίησης χρησιμοποιούν την πληροφορία της κλίσης (gradient) . Γνωστό παράδειγμα gradient-based αλγόριθμου είναι ο Newton-Raphson, καθώς χρησιμοποιεί τις τιμές της συνάρτησης και τις παραγώγους της. Ένας τέτοιος αλγόριθμος λειτουργεί πολύ αποτελεσματικά για ομαλά μονοτροπικά προβλήματα, ωστόσο σε περίπτωση που υπάρχει ασυνέχεια στην αντικειμενική συνάρτηση δεν είναι αποδοτικός και αντ΄ αυτού προτιμάτε ένας non-gradient αλγόριθμος που δεν χρειάζεται κάποια παράγωγο αλλά μόνο τις τιμές της συνάρτησης. Παραδείγματα gradient-free αλγορίθμων είναι ο Hooke-jeeves pattern search και η μέθοδος Nelder-Mead (downhill simplex method).

Οι στοχαστικοί αλγόριθμοι, δηλαδή οι μη-ντετερμινιστικοί, είναι δυο ειδών, οι ευρετικοί και οι μεθευρετικοι. Οι πρώτοι, ψάχνουν την βέλτιστη λύση μέσω της

“δοκιμής και του σφάλματος” (trial & error) και ως εκ τούτου μπορούν να βρουν μια αρκετά καλή λύση σε λογικό χρόνο, χωρίς όμως να μας εγγυούνται ότι είναι η βέλτιστη. Αυτό διότι είναι αδύνατον σε περίπλοκα προβλήματα να εξετάσει όλες τις εφικτές λύσεις σε λογικό χρόνο.

Εξέλιξη των ευρετικών αλγορίθμων αποτελούν οι μεθευρετικοί οι οποίοι είναι πιο αποτελεσματικοί και η βασική διαφορά τους είναι ότι συνδυάζουν την τοπική αναζήτηση (local search) βέλτιστης λύσης με κάποια γενικότερη στρατηγική αναζήτησης ώστε να ξεφύγουν από το τοπικό βέλτιστο (local optimum) και να προσδιορίσουν το ολικό (global optimum). Κοινό στοιχείο ευρετικών και μεθευρετικών αλγορίθμων είναι ότι σε αντίθεση με τους αιτιωκρατικούς δεν μπορούν να εγγυηθούν ότι η λύση που βρέθηκε είναι η βέλτιστη.

### 3.1.2 Μεθευρετικοί Αλγόριθμοι

Τα δυο κύρια συστατικά κάθε μεθευρετικού αλγορίθμου είναι : η επίταση (intensification) και η διαφοροποίηση(diversification) ή εκμετάλλευση (exploitation) και εξερεύνηση (exploration). Η επίταση είναι η εστίαση στην τοπική αναζήτηση εκμεταλλευόμενοι την πληροφορία ότι βρίσκουμε μια καλή λύση σε αυτήν την περιοχή. Αντιθέτως η διαφοροποίηση σημαίνει η εξερεύνηση του χώρου των λύσεων προς αναζήτηση νέας βέλτιστης λύσης. Η διαφοροποίηση εξασφαλίζει ότι ο αλγόριθμος δεν θα παγιδευτεί σε ένα τοπικό ελάχιστο. Με τη σωστή αναλογία των δύο παραπάνω στοιχείων του αλγορίθμου εξασφαλίζεται η εύρεση μιας πολύ καλής λύσης. Όλοι οι αλγόριθμοι της οικογένειας των μεθευρετικών χρησιμοποιούν μεθόδους προκειμένου να συνεχίσουν να ψάχνουν για λύσεις κοντά σε προηγούμενες (exploitation) ή να μη δώσουν μεγάλη βαρύτητα σε ήδη δοκιμασμένες λύσεις (exploration).

Οι μεθευρετικοί αλγόριθμοι μπορούν να κατηγοριοποιηθούν με πολλούς τρόπους . Ένας από αυτούς είναι οι αλγόριθμοι που χρησιμοποιούν πληθυσμούς (population-based) και εκείνοι που χρησιμοποιούν την τροχιά (trajectory based) . Παράδειγμα του πρώτου είδους είναι η μέθοδος βελτιστοποίησης σμήνους (particle swarm optimization) χρησιμοποιώντας έναν πληθυσμό υποψηφίων λύσεων που τον μετακινεί στο χώρο αναζήτησης. Όλοι οι αλγόριθμοι αυτής της ομάδας διαθέτουν εργαλεία που επιτρέπουν την ανταλλαγή πληροφοριών μεταξύ των ατόμων του πληθυσμού προκειμένου να αποφύγουν κακές λύσεις. Αντίθετα ένας αλγόριθμος βασισμένος στην τροχιά, είναι η προσημειωμένη ανόπτηση (simulated annealing), που χρησιμοποιεί ένα στοιχείο –υποψήφια λύση – την οποία μετακινεί τμηματικά στον χώρο των λύσεων. Μια μετακίνηση σε καλύτερη λύση γίνεται πάντα αποδεκτή, ενώ μια μέτρια μετακίνηση γίνεται αποδεκτή με κάποια πιθανότητα.

### 3.1.3 Αλγόριθμοι εμπνευσμένοι από τη φύση

Στη βιβλιογραφία υπάρχει πληθώρα αλγορίθμων που βασίζονται σε διαδικασίες που συμβαίνουν στη φύση και εμπνέονται από επιστήμες όπως η χημεία, η φυσική και κυρίως η βιολογία. Τέτοιοι είναι :

• **Οι Εξελικτικές Στρατηγικές** (Evolutionary Strategies) και οι **Γενετικοί Αλγόριθμοι** (Genetic Algorithms), που προέρχονται από την μίμηση της διαδικασίας της εξέλιξης των ειδών στη φύση. Κατ' αναλογία με τη θεωρία του Δαρβίνου της φυσικής διαλογής, στην οποία επιβιώνει ο ισχυρότερος, έτσι και στην επίλυση του προβλήματος επιδιώκουμε την επικράτηση της βέλτιστης λύσης. Οι δύο αυτοί αλγόριθμοι βασίζονται στο ίδιο σκεπτικό και διαφοροποιούνται στον τρόπο με τον οποίο γίνεται η διαλογή των ατόμων-λύσεων, η διασταύρωση και η μετάλλαξη τους. Η διαλογή αποτελεί το μηχανισμό επίτασης (intensification) του αλγορίθμου, ενώ η διασταύρωση και η μετάλλαξη καθορίζουν τη διαφοροποίηση (diversification).

• **Τα Νευρωνικά Δίκτυα** (Neural Nets ) προέρχονται από τον τρόπο λειτουργίας του ανθρωπίνου εγκεφάλου .Ο εγκέφαλος αποτελείται νευρώνες (10.000.000.000 κατά προσέγγιση), οι οποίοι είναι μαζικά διασυνδεδεμένοι κατά μέσο όρο από διάφορες χιλιάδες διασυνδέσεις ανά νευρώνα. Κάθε νευρώνας είναι ένα εξειδικευμένο κύτταρο το οποίο έχει τη δυνατότητα μετάδοσης ενός ηλεκτροχημικού σήματος. Ένας νευρώνας πυροδοτεί ένα ηλεκτροχημικό σήμα όταν το συνολικό σήμα το οποίο λήφθηκε στο ίδιο, υπερβεί ένα συγκεκριμένο επίπεδο, δηλαδή, το κατώτατο όριο βολής (firing threshold).

• **Η Προσομοιωμένη Ανόπτηση** (Simulated Annealing) που πρώτος χρησιμοποίησε ο kirkpatrick, βασίζεται στην διαδικασία ανόπτησης των μετάλλων, δηλαδή στην θέρμανση τους σε υψηλές θερμοκρασίες και στη συνέχεια η αργή ψύξη τους σε θερμοκρασία δωματίου. Έτσι στην βελτιστοποίηση κάθε λύση αποτελεί μια ενεργειακή στάθμη και ο αλγόριθμος επιδιώκει να βρεθεί στην χαμηλότερη ενεργειακή στάθμη δηλαδή τη βέλτιστη λύση.

• **Ο Αλγόριθμος βελτιστοποίησης Σμήνους Σωματιδίων** (particle swarm optimization) προσομοιώνει την κοινωνική συμπεριφορά ορισμένων οργανισμών στη φύση, όπως τα σμηνοι πουλιών και τα κοπάδια ψαριών. Χαρακτηριστικό του αλγορίθμου είναι η ύπαρξη μνήμης που κληρονομείτε από γενιά σε γενιά. Αυτό έχει σαν πλεονέκτημα ότι λύσεις που έχουν απορριφθεί δεν επαναλαμβάνονται.

• **Ο Αλγόριθμος βελτιστοποίησης Ζευγαρώματος Μελισσών** (Honey Bees mating optimization) είναι ένας σύγχρονος αλγόριθμος που αντιγράφει τη διαδικασία με την οποία η βασίλισσα κάνει την τυχαία πτήση ζευγαρώματος, ακολουθούμενη από το σμήνος των κηφήνων και επιλέγει με βάση την ταχύτητα της, τον κηφήνα με τον οποίο θα ζευγαρώσει.

• **Ο Αλγόριθμος της Αποικίας Μυρμηγκιών** (ant colony optimization), όπως διαφαίνεται και από το όνομα του, είναι εμπνευσμένος από την συμπεριφορά των μυρμηγκιών στη φύση και συγκεκριμένα από τον τρόπο με τον οποίο αυτά αναζητούν το βέλτιστο μονοπάτι μεταξύ της αποικίας και της πηγής της τροφής. Στο επόμενο τμήμα της εργασίας περιγράφεται αναλυτικά η διαδικασία που ακολουθούνε τα μυρμήγκια.

## 3.2 Κοινωνικά έντομα

### 3.2.1 Αυτο-οργάνωση

Αυτό-οργάνωση είναι η διαδικασία όπου κάποιας μορφής τάξη ανακύπτει στο "γενικό επίπεδο", μέσω των αλληλεπιδράσεων ανάμεσα στα στοιχεία που βρίσκονται στο "τοπικό επίπεδο", ενός αρχικά ανοργάνωτου συστήματος.

Αυτή η διαδικασία είναι αυθόρμητη, δηλαδή δεν ελέγχεται, ούτε καθοδηγείται από οποιοδήποτε στοιχείο εντός του συστήματος αλλά ούτε και εκτός αυτού. Παρόλα αυτά οι κανόνες που διέπουν το σύστημα και οι αρχικές του συνθήκες μπορεί να έχουν επιλεγεί η προκληθεί από κάποιον εντός ή εκτός του συστήματος.

Η αυτό-οργάνωση ως έννοια , συναντάται σε πληθώρα θετικών επιστημών όπως η φυσική , η χημεία, τα μαθηματικά , η οικονομία , η πληροφορική αλλά και σε πολλές κοινωνικές επιστήμες. Στην εργασία αυτή , θα αναλύσουμε την αυτο-οργάνωση από τη σκοπιά της βιολογίας και συγκεκριμένα , πως αυτή λαμβάνει χώρα σε μια αποικία μυρμηγκιών.

Ένα αυτο-οργανωμένο σύστημα , περιλαμβάνει τα εξής χαρακτηριστικά.

➢ Την **Θετική Ανατροφοδότηση** .(Την ενίσχυση). Δηλαδή την συμπεριφορά που ενισχύει τη δημιουργία μιας δομής. Στην περίπτωση των μυρμηγκιών και συγκεκριμένα στην αναζήτηση της τροφής τους, ένα παράδειγμα ενίσχυσης είναι η προτροπή με κάποιον τρόπο και άλλων μυρμηγκιών να ακολουθήσουν το ίδιο μονοπάτι όπου ένα μυρμήγκι βρήκε τροφή.

➢ Την **Αρνητική Ανατροφοδότηση**. (απόσβεση) . Ουσιαστικά το αντίβαρο της ενίσχυσης, την αποθάρρυνση από τη δημιουργία της συγκεκριμένης δομής. Παράδειγμα απόσβεσης στον πληθυσμό των μυρμηγκιών είναι η περίπτωση όπου μυρμήγκι ειδοποιεί τα υπόλοιπα ότι τα αποθέματα της τροφής στην συγκεκριμένη πηγή εξαντλούνται, μειώνονται ή υπάρχει συνωστισμός.

➢ Τις **Διακυμάνσεις**. Με άλλα λόγια ο βαθμός τυχαιότητας με την οποία δημιουργούνται οι δομές. Συγκεκριμένα στην αναζήτηση τροφής των μυρμηγκιών, το κατά πόσο θα ακολουθήσει ένα μυργμήγκι την προτροπή ενός άλλου να ακολουθήσει συγκεκριμένο μονοπάτι, παίζει τεράστιο ρόλο στην αποτελεσματικότητα της αναζήτησης, καθότι χωρίς την ύπαρξη οποιασδήποτε τυχαιότητας θα περιοριζόντουσαν μόνο στην πηγή τροφής που θα έβρισκε το πρώτο μυρμήγκι.

Επομένως τα “λάθη” και οι αποκλίσεις στις διαδρομές δεν αποτελούν απαραίτητα κάτι αρνητικό.

➢ Τις **Αλληλεπιδράσεις**. Το κάθε άτομο από μόνο του, μπορεί να δημιουργήσει μια δομή, ωστόσο σε ένα αυτοοργανομένο σύστημα οφείλει να ανταλλάσσει πληροφορίες με τον υπόλοιπο πληθυσμό. Δηλαδή να επηρεάζει με το έργο του και να επηρεάζεται από το έργο των άλλων.

### 3.2.2 Στιγμεργία

Στην προηγούμενη παράγραφο αναφερθήκαμε στην ανάγκη ύπαρξης αλληλεπίδρασης μεταξύ των μυρμηγκιών, δηλαδή κάποιο τρόπο επικοινωνίας. Αυτή μπορεί να είναι άμεση ή έμμεση. Η άμεση επικοινωνία συνίσταται στην ανταλλαγή της πληροφορίας μέσω την οπτικής επαφής των εντόμων, ή με την ανταλλαγή χημικών ουσιών μεταξύ τους. Η έμμεση επικοινωνία συνίσταται στην πληροφορία που μεταβιβάζουν μεταβάλλοντας το περιβάλλον τους το οποίο κατ' επέκταση συνδιαμορφώνουν με τα άλλα μέλη του πληθυσμού.

Πρωτοπόρος στην μελέτη της κοινωνικής συμπεριφοράς των εντόμων ήταν τη δεκαετία του 1940 ο Γάλλος εντομολόγος Pierre-Paul Grasse. Το 1946 ανακάλυψε οτι τα έντομα ήταν ικανά να αντιδρούν κατά ένα συγκεκριμένο τρόπο, γενετικά καθορισμένο, σε ερεθίσματα στο περιβάλλον τους . Αργότερα, το 1959, διαπίστωσε οτι τα αποτελέσματα αυτών των αντιδράσεων λειτουργούν ως νέα ερεθίσματα τόσο για τα έντομα που τα παρήγαγαν όσο και για τα υπόλοιπα.

Ο όρος που χρησιμοποίησε για να περιγράψει “το ερέθισμα που προκαλούν οι εργάτες (κοινωνική τάξη των τερμιτών) στο περιβάλλον τους, ως αποτέλεσμα της εργασίας τους” είναι η στιγμεργία .Αυτό που διαφοροποιεί την στιγμεργία από άλλους τρόπους επικοινωνίας είναι :

- Η απτή, μη συμβολική φύση της πληροφορίας όπου αντιστοιχεί στην αλλαγή που επιφέρει το έντομο στο περιβάλλων που επισκέπτεται.
- Η τοπική φύση της πληροφορίας, όπου πρόσβαση έχουν μόνο τα έντομα που θα βρεθούν στον τόπο που εκείνη έχει εναποθετηθεί.

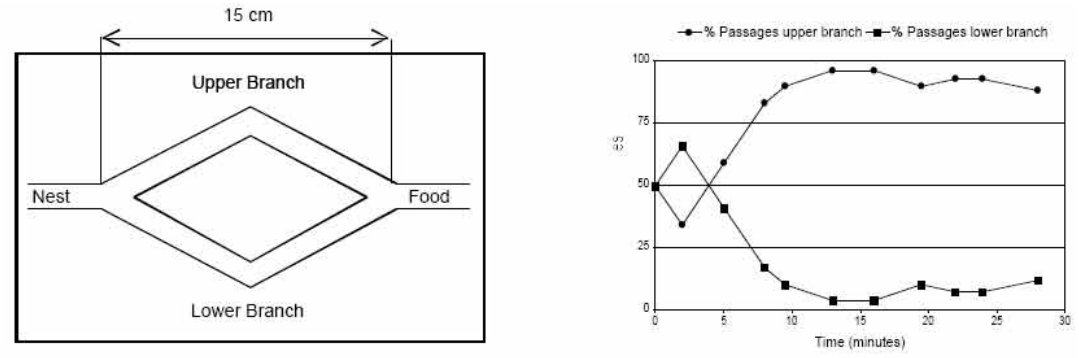
Η στιγμεργία παρατηρείται και στις αποικίες των μυρμηγκιών. Σε πολλά είδη μυρμηγκιών, καθώς αυτά περπατάνε από και προς την πηγή τροφής, εναποθέτουν στο έδαφος την ουσία φερομόνη. Τα υπόλοιπα μυρμήγκια, οσμίζονται την φερομόνη και η παρουσία της επηρεάζει την απόφαση τους για τον ποιο δρόμο θα ακολουθήσουν.

### 3.2.3 Αναζήτηση τροφής

Τα μυρμήγκια δημιουργούν μονοπάτια φερομόνης κατά τη διάρκεια αναζήτησης της πηγής της τροφής και επιστρέφουν στη φωλιά αφήνοντας και πάλι το στίγμα τους στη διαδρομή που ακολουθούν. Τα υπόλοιπα μυρμήγκια εντοπίζουν αυτά τα μονοπάτια και ανάλογα την ένταση της φερομόνης σε αυτά, αποφασίζουν με κάποια πιθανότητα αν θα τα ακολουθήσουν ή όχι.

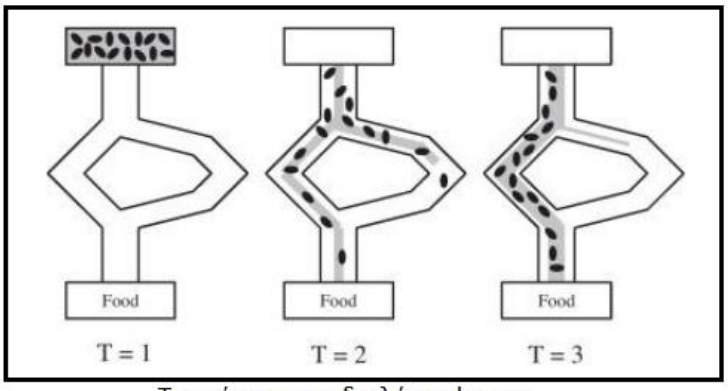
Το 1990 ο deneubourg έκανε ένα πείραμα ονόματι "το πείραμα της δυαδικής γέφυρας" (binary bridge experiment),προκειμένου να μελετήσει την συμπεριφορά των μυρμηγκιών και να ποσοτικοποιήσει την αλληλεπίδραση τους με τη φερομόνη. Χρησιμοποιώντας μυρμήγκια Linepithema humile, συνέδεσε την φωλιά των μυρμηγκιών με την πηγή της τροφής, μέσω 2 γεφυρών ίσου μήκους. Τα μυρμήγκια ελεύθερα μπορούσαν να επιλέξουν από ποια γέφυρα θα έψαχναν για την τροφή.

Αρχικά οι γέφυρες δεν είχαν φερομόνη. Τα μυρμήγκια εξερευνούν τον χώρο και τελικά περνάνε μία από τις γέφυρες. Καθώς κινούνται από και προς την τροφή, εναποθέτουν φερομόνη στη γέφυρα που χρησιμοποιούν. Με το πέρας του χρόνου, λόγο της διακύμανσης, τα επίπεδα φερομόνης στη μία γέφυρα θα είναι ισχυρότερα. Τα μυρμήγκια τείνουν να ακολουθήσουν το μονοπάτι με την εντονότερη φερομένη, επομένως η αποικία μυρμηγκιών συγκλίνει στην χρησιμοποίηση μιας μόνο γέφυρας.




Το πειραμα δυαδικής γέφυρας του deneubourg et al., 1990

Σε ένα άλλο πείραμα το 1989,ο Goss, χρησιμοποίησε δυο γέφυρες διαφορετικού μήκους. Παρατήρησε ότι η τυχαία διακύμανση στην αρχική επιλογή γέφυρας πέρασε σε δεύτερη μοίρα, καθώς, τα μυρμήγκια που κατά τύχη διάλεγαν την συντομότερη γέφυρα για να φτάσουν στην τροφή, γυρνούσαν στη φωλιά από την ίδια με μεγαλύτερη πιθανότητα. Έτσι χάρις την φερομόνη σύντομα όλα τα μυρμήγκια συνέκλιναν στην χρήση της συντομότερης διαδρομής.



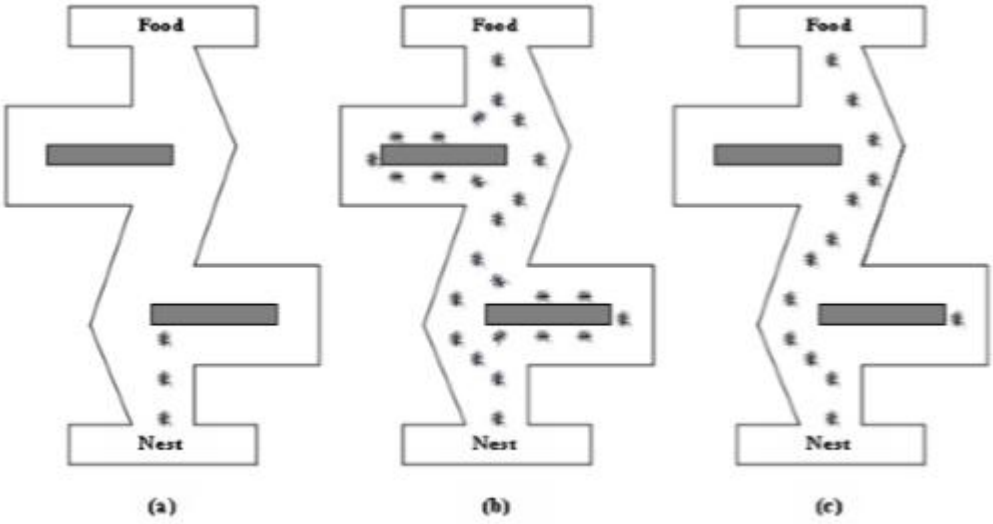
Το πείραμα της διπλής γεφυρας

Αναλυτικότερα η διαδικασία εύρεσης τροφής των μυρμηγκιών είναι η εξής:

1. Τα μυρμήγκια ξεκινούν την αναζήτηση τροφής γύρο από τη περιοχή της πηγής της τροφής κατά τυχαίο τρόπο, επιδιώκοντας της εύρεση της συντομότερης διαδρομής.
2. Κατά τη διάρκεια της περιπλάνησης τους εκκρίνουν τη φερομόνη στο μονοπάτι που ακολουθούν. Η ποσότητα της φερομόνης εξαρτάται από την απόσταση, την ποιότητα και την ποσότητα της τροφής που υπάρχει στην πηγή.
3. Το επόμενο μυρμήγκι που θα ξεκινήσει την αναζήτηση τροφής, επιλέγει ή όχι με κάποια πιθανότητα, το μονοπάτι που ήδη έχει φερομόνη.
4. Το μυρμήγκι θα αφήσει και αυτό με τη σειρά του το αποτύπωμα φερομόνης στο μονοπάτι που θα ακολουθήσει. Αν επιλέξει ένα ήδη μαρκαρισμένο μονοπάτι, τότε αυτό θα ενισχυθεί. Καθώς η ώρα περνάει, η φερομόνη εξατμίζεται και επομένως τα μονοπάτια που δεν είναι δελεαστικά για τα μυρμήγκια εξαλείφονται.
5. Εν τέλη, από όλα τα αρχικά μονοπάτια, υπερισχύει ένα μόνο, αυτό της συντομότερης διαδρομής.

Ορισμένα είδη μυρμηγκιών καθώς πηγαίνουν προς την πηγή της τροφής από ένα νέο μονοπάτι (χωρίς φερομόνη), εναποθέτουν τη φερομόνη όχι συνεχόμενα, αλλά σε συγκεκριμένες αποστάσεις. Κατά την επιστροφή τους στη φωλιά, αν ακολουθήσουν το μονοπάτι από το οποίο ήρθαν, θα εναποθέσουν φερομόνη συνεχόμενα και σύμφωνα με τα κριτήρια που προαναφέρθηκαν. Επίσης όταν το πρώτο μυρμήγκι εντοπίσει τροφή, κρατάει μια ποσότητα στο στόμα του και όταν επιστρέφει στην φωλιά ταΐζει μερικά από τον υπόλοιπο πληθυσμό ώστε να τα "ειδοποιήσει" ότι το μονοπάτι που δημιούργησε, οδηγεί σε τροφή.

Από αυτήν την διαδικασία και κατά αναλογία με τον τρόπο που τα μυρμήγκια βρίσκουν την συντομότερη διαδρομή, δημιουργήθηκαν αλγόριθμοι που στόχο έχουν την εύρεση της βέλτιστης λύσης σε πολλές κατηγορίες προβλημάτων. Στην παρακάτω εικόνα γίνεται ευκολότερα κατανοητή η διαδικασία της αναζήτησης τροφής. Στο σχέδιο (a) τα μυρμήγκια ξεκινάει τυχαία η διαδρομή τους, στο (b) υπάρχει ποικιλία λύσεων, ενώ στο (c) πλέον τα περισσότερα έχουν συγκλίνει σε μια λύση.




## 3.3 Ο αλγόριθμος Ant System-Ranked

### 3.3.1 Ιστορική αναδρομή

Ο πρώτος αλγόριθμος βασισμένος στη διαδικασία εύρεσης τροφής των μυρμηγκιών παρουσιάστηκε το 1992 στην διδακτορική διατριβή του M.Dorigo το 1992 στο Πολυτεχνείο του Μιλάνου και ονομαζόταν Ant System. Ο αλγόριθμος αυτός, εμπνευσμένος από τις παρατηρήσεις του Deneubourg και του Goss πάνω στα κοινωνικά έντομα και την έμμεση επικοινωνία των μυρμηγκιών με την μέθοδο της στιγμεργιας για την αναζήτηση τροφής, στα επόμενα χρόνια εξελίχθηκε σε αυτό που σήμερα είναι γνωστό ως Ant Colony Optimization.

Ο αλγόριθμος Ant System επιλύει διακριτά προβλήματα, όπως για παράδειγμα το πρόβλημα του πλανόδιου πωλητή (TSP), με πάρα πολύ καλά αποτελέσματα. Αργότερα υπήρξαν πολλές παραλλαγές του, όπως ο Elitist Ant System όπου η καλύτερη λύση ενισχύεται σε κάθε επανάληψη, με αποτέλεσμα την ταχύτερη σύγκλιση. Μια άλλη παραλλαγή είναι ο Max-Min Ant System, όπου όπως προδίδει και η ονομασία του, τα επίπεδα φερομόνης κινούνται μεταξύ 2 τιμών. Άλλες τροποποιήσεις είναι οι αλγόριθμοι Rank-based Ant System, Continuous Orthogonal Ant Colony και ACO with Fuzzy Logic με τον τελευταίο να συνδυάζει τον αλγόριθμο των μυρμηγκιών με την ασαφή λογική προκειμένου τα μυρμήγκια να είναι αποτελεσματικότερα στην αναζήτηση των μονοπατιών που θα ακολουθήσουν.

### 3.3.2 Η "ψηφιοποίηση" της διαδικασίας

Η μετατροπή της φυσικής διαδικασίας εύρεσης της βέλτιστης διαδρομής σε αλγόριθμο υπολογισμού της σε κάποιο ψηφιακό μοντέλο, προϋποθέτει ορισμένες τροποποιήσεις. Τα μυρμήγκια που χρησιμοποιεί ο αλγόριθμος θα είναι απλουστευμένα σε σχέση με τα πραγματικά ωστόσο έχουν επιπλέον ιδιότητες, ανάλογα το πρόβλημα που επιλύουν.

Οι αντιστοιχίες φυσικών ιδιοτήτων με τις ψηφιακές είναι οι εξής :

- Ο αλγόριθμος χρησιμοποιεί ένα πληθυσμό από **μυρμήγκια** όπου κάθε ένα από αυτά θα ακολουθήσει ένα μονοπάτι το οποίο είναι μια λύση του προβλήματος. Ο πληθυσμός των μυρμηγκιών είναι μια σημαντική παράμετρος του αλγορίθμου που επηρεάζει το χρόνο προσέγγισης της βέλτιστης λύσης.
- Η **φερομόνη** στον αλγόριθμο βελτιστοποίησης της Αποικίας Μυρμηγκιών αντιστοιχεί σε μια αριθμητική πληροφορία που θα εναποθέτουν τα μυρμήγκια και θα είναι προσβάσιμη από όλα τα υπόλοιπα, σχετικά με τη διαδρομή που ακολούθησαν. Αυτή η πληροφορία θα πρέπει να φθίνει στο χρόνο με τρόπο αντίστοιχο της εξάτμισης της φερομόνης στη φύση.
- Η **Επιλογή** της διαδρομής που θα ακολουθήσουν δεν είναι καθαρά αιτιοκρατικός, αλλά είναι σε ένα βαθμό τυχαίος. Τα ψηφιακά μυρμήγκια δεν έχουν μνήμη και επομένως η επιλογή της διαδρομής που θα ακολουθήσουν (στο αιτιοκρατικό της μέρος), εξαρτάται μόνο από την πληροφορία που λαμβάνει από στο συγκεκριμένο χώρο, τον συγκεκριμένο χρόνο και δεν γνωρίζει τι συνέβαινε σε παρελθόντα χρόνο ούτε τη πληροφορία υπάρχει στον υπόλοιπο χώρο.

Τα ψηφιακά μυρμήγκια εκτός από τα βασικά χαρακτηριστικά τους, μπορούν ανάλογα με την εφαρμογή να εφοδιαστούν με επιπλέον ιδιότητες, που θα επηρεάζουν την ταχύτητα σύγκλισης στη βέλτιστη λύση και γενικά την απόδοση του αλγορίθμου.

Τα μυρμήγκια του αλγορίθμου μπορούν να αποκτήσουν μνήμη των προηγούμενων πράξεων τους και μονοπατιών που έχουν διασχίσει. Επίσης η ύπαρξη όρασης στο χώρο πέραν του σημείου που βρίσκονται, και ο υπολογισμός των αποστάσεων προτού διασχίσουν μια διαδρομή. Τα ψηφιακά μυρμήγκια έχουν την ικανότητα να εναποθέτουν τη φερομόνη κατά τη διάρκεια της περιπλάνησης τους ή μετά το πέρας αυτής. Ένα επιπλέον προσόν είναι η ικανότητα να χρησιμοποιούν αλγορίθμους ώστε να βελτιστοποιούν τοπικά τη διαδρομή.

### 3.3.3 Εφαρμογές του ACO

Αλγόριθμοι ACO εφαρμόζονται σε πληθώρα προβλημάτων συνδυαστικής βελτιστοποίησης και έχουν επεκταθεί στην επίλυση δυναμικών προβλημάτων, στοχαστικών προβλημάτων και προβλημάτων όπου χρησιμοποιούνται

συμπληρωματικά. Έχουν σημαντικό πλεονέκτημα έναντι άλλων αλγορίθμων όπως για παράδειγμα της Προσομοιωμένης Ανόπτησης και των γενετικών αλγορίθμων όσο αφορά προβλήματα που αλλάζουν δυναμικά, διότι η αποικία μυρμηγκιών προσαρμόζεται σε αλλαγές σε πραγματικό χρόνο. Για το λόγο αυτό βρίσκει πολλές εφαρμογές σε δίκτυα αστικών συγκοινωνιών. Ο πρώτος αλγόριθμος βελτιστοποίησης που χρησιμοποιεί την αποικία μυρμηγκιών είναι ο Ant System. Ο στόχος του αρχικά ήταν η επίλυση του προβλήματος του πλανόδιου πωλητή (TSP) ωστόσο η εξέλιξη του, οι αλγόριθμοι ACO επιλύουν μια τεράστια γκάμα προβλημάτων πολλών κατηγοριών. Είναι πολύ ισχυρός σε διακριτά προβλήματα ενώ σε συνεχή η απόδοση του δεν θεωρείται ιδιαίτερα ικανοποιητική. Γενικότερα, προβλήματα που μπορούν να μοντελοποιηθούν σε ένα γράφημα κόμβων και τόξων, μπορούν να επιλυθούν από αλγορίθμους ACO ικανοποιητικά. Ενδεικτικά αναφέρονται:

Προβλήματα Χρονοπρογραμματισμού - Scheduling Problems

- Πρόβλημα προγραμματισμού & ελέγχου παραγωγής σε σταθμό εργασίας, Job-shop scheduling problem (JSP)
- Ανοιχτό πρόβλημα προγραμματισμού & ελέγχου παραγωγής, Open-shop scheduling problem (OSP)
- Μεταθετικό πρόβλημα προγραμματισμού γραμμής παραγωγής, flow shop problem (PFSP)
- Πρόβλημα ολικής βραδύτητας μιας μηχανής,Single machine total tardiness problem (SMTTP)
- Προγραμματισμός εργασιών με περιορισμούς πόρων, Resource-constrained project scheduling problem (RCPSP)
- Πρόβλημα προγραμματισμού παραγωγής σε πολλαπλούς σταθμούς εργασίας, Group-shop scheduling problem (GSP)

Προβλήματα Δρομολόγησης Οχημάτων (ΠΔΟ) -Vehicle Routing Problems

- ΠΔΟ με περιορισμό χωρητικότητας, Capacitated vehicle routing problem (CVRP)
- ΠΔΟ, με πολλαπλές αποθήκες Multi-depot vehicle routing problem (MDVRP)
- Περιοδικό ΠΔΟ,Period vehicle routing problem (PVRP)
- Στοχαστικό ΠΔΟ,Stochastic vehicle routing problem (SVRP)
- ΠΔΟ με παραλαβή και διανομή Vehicle routing problem with pick-up and delivery(VRPPD)
- ΠΔΟ με χρονικά παράθυρα,Vehicle routing problem with time windows (VRPTW)
- Χρονικά εξαρτώμενο ΠΔΟ με χρονικά παράθυρα, Time Dependent Vehicle Routing Problem with Time Windows (TDVRPTW)

Προβλήματα Ανάθεσης - Assignment Problem

- Πρόβλημα τετραγωνικής ανάθεσης,Quadratic assignment problem(QAP)
- Γενικό πρόβλημα ανάθεσης,Generalized assignment problem(GAP)
- Πρόβλημα ανάθεσης με συχνότητες,Frequency assignment problem(FAP)
- Πρόβλημα κατανομής εφεδρείας, Redundancy allocation problem(RAP)

Επίσης χρησιμοποιούνται σε εφαρμογές Classification, Data Minning, Image processing και πολλές.

### 3.3.4 Μαθηματικές σχέσεις του ACO για το πρόβλημα της δρομολόγησης

Στα προβλήματα δρομολόγησης οχημάτων (VRP), το ζητούμενο είναι η εύρεση της βέλτιστης οικονομικά, χρονικά ή χωρικά διαδρομής που πρέπει να ακολουθήσουν οχήματα με συγκεκριμένη αφετηρία (και κατάληξη) προκειμένου να ικανοποιήσουν μια δεδομένη ζήτηση, τηρώντας ορισμένους περιορισμούς.

Σε αυτού του είδους τα προβλήματα η μαθηματική εφαρμογή του ACO γίνεται ως εξής [11]:

Το μυρμήγκι-λύση έχοντας ως αφετηρία μια πόλη/αποθήκη i αποφασίζει να κινηθεί προς κάποια πόλη j ώστε να διανείμει μια ποσότητα προϊόντος. Προκειμένου να διαλέξει ποια θα είναι αυτή, χρησιμοποιεί 2 πληροφορίες. Η μια πληροφορία είναι η ευρετική πληροφορία $n_{ij}=1/c_{ij}$ όπου εκφράζει το πόσο θελκτική είναι η επιλογή αυτής διαδρομής ως συνάρτηση αντιστρόφως ανάλογης του κόστους (ή του χρόνου ή της απόστασης) $c_{ij}$ μεταξύ αφετηρίας και προορισμού. Η δεύτερη πληροφορία είναι τα επίπεδα φερομόνης $\tau_{ij}$ που υπάρχουν στην συγκεκριμένη διαδρομή. Η ποσότητα $\tau_{ij}$ που εντοπίζει το μυρμήγκι προέρχεται από την φερομόνη που έχουν αφήσει τα μυρμήγκια που πέρασαν από εκεί σε προηγούμενες χρονικές στιγμές. Όσο περισσότερη, τόσο πιο συμφέρουσα την θεώρησαν τα υπόλοιπα μυρμήγκια.

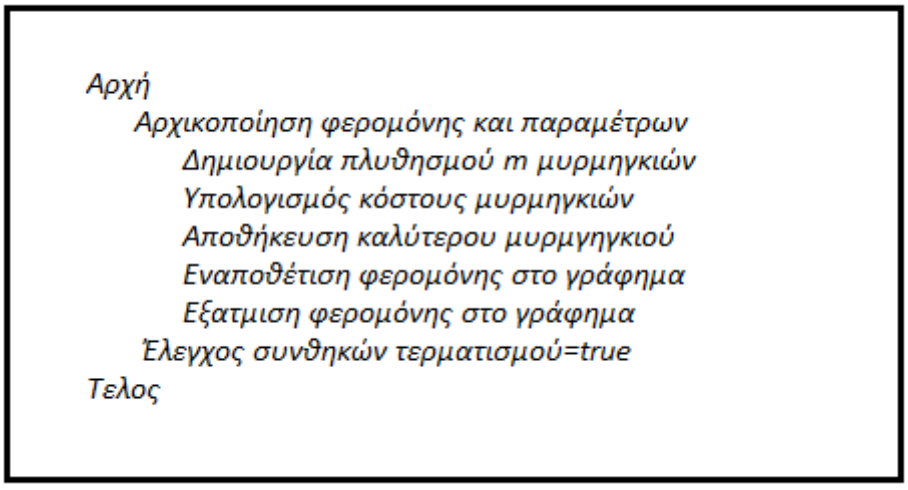
Έχοντας αυτά τα δυο δεδομένα το μυρμήγκι-λύση ακολουθεί τη διαδρομή (i,j) με πιθανότητα ίση με $p_{ij} = \frac{[\tau_{ij}]^a[n_{ij}]^b}{\sum_{l=1}^{M}[\tau_{lj}]^a[n_{lj}]^b}$, όπου M το σύνολο των πόλεων που μπορεί να μεταβεί το μυρμήγκι μετά τον κόμβο i και *a*, *b* παράμετροι που καθορίζουν το βαθμό τυχαιότητα στην επιλογή της διαδρομής. Αν το α=0 τότε η απόφαση δεν λαμβάνει υπόψη τη φερομόνη που υπάρχει στο μονοπάτι αυτό. Αντιθέτως αν b=0 τότε η απόφαση επαφίεται καθαρά στην πληροφορία για το πόσο καλή είναι η διαδρομή αυτή με βάση τα προηγούμενα μυρμήγκια.

Όπως συμβαίνει και στη φύση, όπου η φερομόνη εξατμίζεται με το πέρασμα του χρόνου έτσι και στον αλγόριθμο αυτό συμβαίνει σε κάθε επανάληψη, δηλαδή όταν όλα τα μυρμήγκια έχουν ολοκληρώσει μια λύση του προβλήματος. Ο

συντελεστής εξάτμισης συμβολίζεται με ρ και εφαρμόζεται σε όλες ή μερικές διαδρομές (ανάλογα την εφαρμογή) σύμφωνα με τον τύπο $\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old}$ με 0<ρ<1. Έτσι οι κακές λύσεις χρησιμοποιούνται όλο και λιγότερο και τείνουν να εξαλείφονται. Η φερομόνη που εναποθέτει κάθε μυρμήγκι δίνεται από τον τύπο $\Delta\tau_{ij} = 1/c$, όπου c το κόστος της λύσης που δημιούργησε και αφορά μόνο τα τόξα από όπου πέρασε. Συνολικά, μετά από κάθε επανάληψη η ανανέωση της φερομόνης γίνεται με την μαθηματική έκφραση $\tau_{ij}^{new} = \tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}$, όπου m ο πληθυσμός των μυρμηγκιών.

Γίνεται αντιληπτό ότι η εφαρμογή του αλγορίθμου δεν είναι απόλυτη αλλά εξαρτάται από το είδος του προβλήματος δρομολόγησης καθώς και τις ιδιομορφίες που κάθε ένα παρουσιάζει. Πάρα ταύτα, το γενικό πλαίσιο στο οποίο όλοι κινούνται είναι αυτό που παρουσιάστηκε παραπάνω. Πέραν των συντελεστών α , b και του αριθμού των μυρμηγκιών m που επαφίενται στην επιλογή του χρήστη υπάρχουν και διαφορές που αφορούν τον τρόπο ανανέωσης της φερομόνης. Για παράδειγμα ο αλγόριθμος Elitist Ant System όπου το βέλτιστο μυρμήγκι εναποθέτει φερομόνη σε κάθε επανάληψη, ο αλγόριθμος Max-Min Ant System όπου η ποσότητα φερομόνης κινείται σε συγκεκριμένο εύρος και ο Rank-based Ant System όπου οι λύσεις ταξινομούνται με βάση το κόστος τους και η φερομόνη που αφήνουν τα μυρμήγκια είναι ανάλογη της θέσης τους σε αυτήν την κατάταξη.

Στο σχήμα που ακολουθεί φαίνεται η γενική μορφή του ACO και η στρατηγική του για την αναζήτηση της βέλτιστης λύσης.



# ΚΕΦΑΛΑΙΟ 4: ΕΦΑΡΜΟΓΗ ΚΑΙ ΕΠΙΛΥΣΗ

## 4.1 Αλγόριθμοι και μέθοδοι που χρησιμοποιήθηκαν

Στην παρούσα εργασία υλοποιήθηκαν πέντε εκδοχές του αλγορίθμου AS-Ranked. Η πρώτη εκδοχή αφορά τον κλασικό AS-Ranked χωρίς τοπική αναζήτηση και χωρίς ρυθμιζόμενη παράμετρο ***a***. Στη δεύτερη εκδοχή προστίθεται η ρυθμιζόμενη παράμετρος ***a***. Οι υπόλοιπες τρεις αφορούν τον ίδιο αλγόριθμο με την προσθήκη τοπικής αναζήτησης. Στόχος είναι αφενός η καλύτερη δυνατή επίλυση των προβλημάτων και αφετέρου η μελέτη της αποτελεσματικότητας της χρήσης μεταβλητής παραμέτρου ***a***. Παρακάτω παρουσιάζονται οι μέθοδοι/αλγόριθμοι που χρησιμοποιούνται στις εκδοχές του κώδικα.

AS-Ranked

Αρχικά παρουσιάζουμε τον κορμό των πέντε εφαρμογών δηλαδή τον αλγόριθμο AS-Rank. Βασίζεται στον απλό Ant System που παρουσιάστηκε στο προηγούμενο κεφάλαιο αποτελεί όμως μια βελτίωση του καθώς περισσότερα μυρμήγκια εναποθέτουν φερομόνη αυξάνοντας το πλάτος της αναζήτησης στο χώρο των λύσεων. Αναλυτικότερα, τα ***m*** μυρμήγκια κάθε επανάληψης αφού ολοκληρώσουν τις διαδρομές τους κοστολογούνται και κατατάσσονται σε αύξουσα σειρά. Από τα ***m*** μυρμήγκια μόνο τα (w-1) καλύτερα αφήνουν φερομόνη στο γράφημα. Η φερομόνη που εναποθέτουν μειώνεται αναλογικά όσο αυξάνεται ο αριθμός κατάταξής τους. Επιπλέον, σε κάθε επανάληψη το μυρμήγκι της καλύτερής έως τώρα λύσης (BS) εναποθέτει φερομόνη πολλαπλασιασμένη με το συντελεστή w. Η μαθηματική έκφραση που περιγράφει αυτή τη διαδικασία είναι:

$$\tau_{ij}^{new} = \tau_{ij}^{old} + \sum_{r=1}^{w-1} (\mathrm{w} - \mathrm{r}) \Delta \tau_{ij}^{r} + w \Delta \tau_{ij}^{BS}$$

Παρακάτω παρουσιάζεται η υλοποίηση του αλγόριθμου AS-Rank για το Συσσωρευτικό Πρόβλημα Δρομολόγησης Οχημάτων σε φυσική γλώσσα:

```
Αρχικοποίησε φερομόνη, ευρετική πληροφορία
ΓΙΑ εναν αριθμό επαναλήψεων
        ΓΙΑ κάθε μυρμήγκι
                -Αρχικοποίησε κόμβους που δεν έχει επισκεφθεί και απόθεμα
                -Θέσε την αποθήκη ως κόμβο που έχει επισκεφθεί
                ΟΣΟ δεν έχει επισκεφθεί όλες τους κόμβους-πόλεις
                        -Υπολόγισε πιθανότητες μετάβασης
                        -Βρές επόμενο κόμβο
                        ΑΝ το απόθεμα αρκεί
                                -Καταχώρισε τη μετάβαση στη λύση
                                -Αφαίρεσε τον κόμβο από τη λίστα
                                -Αφαίρεσε τη ζήτηση του κόμβου από το απόθεμα
                                -Θέσε επόμενο κόμβο αφετηρίας τον παρόντα.
                        ΤΕΛΟΣ_ΑΝ
                        ΑΝ έχεις επισκεφτεί όλους τους κόμβους Ή δεν αρκεί το απόθεμα
                                -Καταχώρησε τη μετάβαση πρός την αποθήκη
                                -Θέσε αφετηρία την αποθήκη
                                -Αρχικοποίησε το απόθεμα
                        ΤΕΛΟΣ_ΑΝ
                ΤΕΛΟΣ_ΟΣΟ
                -Ανέλυσε τη λύση σε δρομολόγια ίσα με τον ελάχιστο αριθμό στόλου
                ΑΝ είναι αδύνατη επανέλαβε το μυρμήγκι
        ΤΕΛΟΣ_ΓΙΑ
        -Εφάρμοσε τοπική αναζήτηση (προαιρετικά)
        -Κατέταξε τα δρομολόγια κατα αυξουσα σειρά κόστους
        -Ανανέωσε τη φερομόνη στο γράφημα
ΤΕΛΟΣ_ΓΙΑ
-Επέστρεψε την καλύτερη λύση
```

Fuzzy Alpha

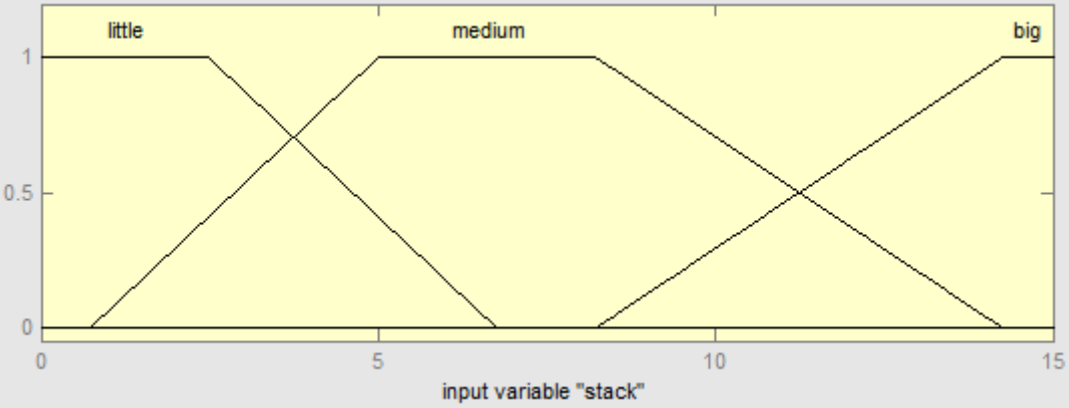
Για τη ρύθμιση της παραμέτρου *α* χρησιμοποιήθηκε ένας fuzzy controller τύπου Mamdani, μιας εισόδου και μιας εξόδου. Ο ελεγκτής είναι τριών κανόνων και ως είσοδο έχει το άθροισμα των φορών που το καλύτερο μυρμήγκι της εκάστοτε επανάληψης ταυτίζεται με το καλύτερο έως τώρα μυρμήγκι. Όταν το καλύτερο μυρμήγκι βελτιώνει τη λύση, το άθροισμα μηδενίζεται ενώ όταν είναι χειρότερη η λύση, το άθροισμα παραμένει το ίδιο. Ως έξοδο ο ελεγκτής δίνει την τιμή της παραμέτρου ***α***. Για το λογικό AND και το Implication χρησιμοποιείται ο τελεστής min, για το λογικό OR και το Aggregation χρησιμοποιείται το MAX. Η μέθοδος που γίνεται η αποασσαφοποίηση (defuzzyfication) είναι η κεντροειδής (Centroid). Οι κανόνες είναι οι εξής:

Έστω η λεκτική μεταβλητή εισόδου Stack και εξόδου Alpha.

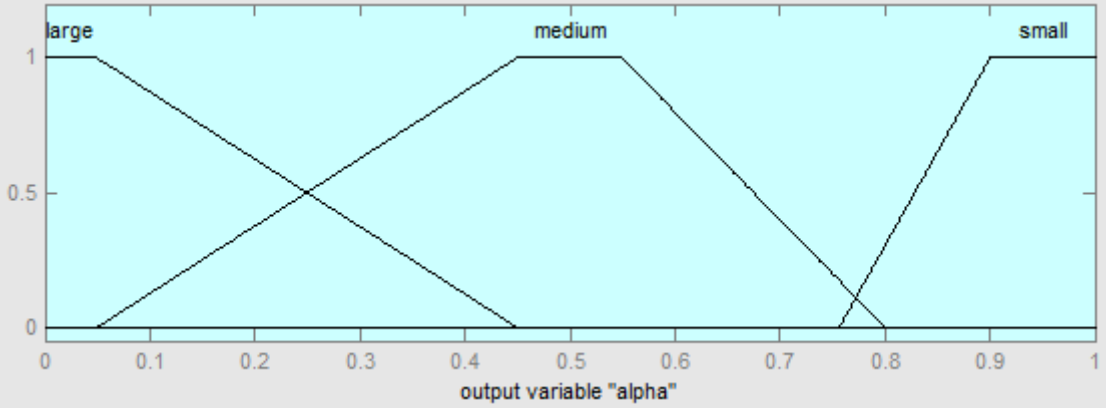
1. IF Stack="little" THEN Alpha ="large"
2. IF Stack="medium" THEN Alpha ="medium"
3. IF Stack="big" THEN Alpha ="small"

Παρακάτω παρουσιάζονται οι συναρτήσεις συμμετοχής (membership functions):

*Μεταβλητή εισόδου*




*Μεταβλητή εξόδου*




Ο μαθηματικός τύπος που περιγράφει τη διαδικασία αποασσαφοποίησης Centroid είναι:

$$a = COG = \frac{\int \mu_{c(x)} \cdot x dx}{\int \mu_{c(x)} dx}$$

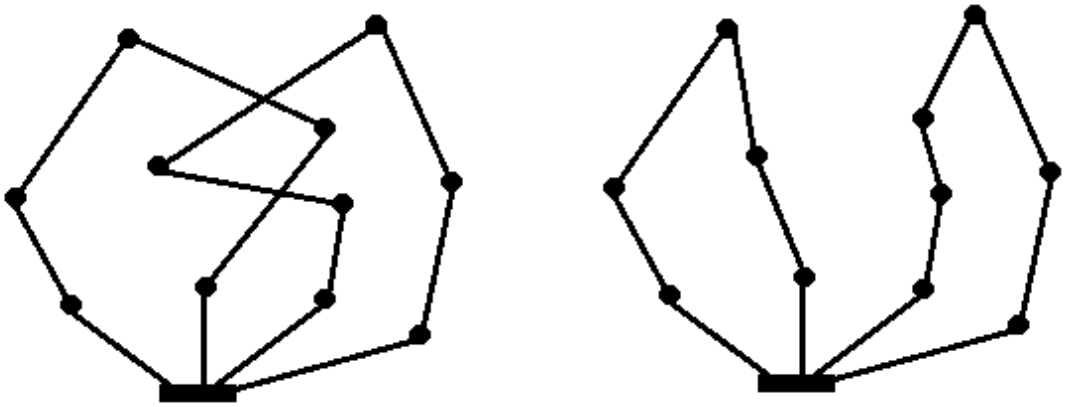
Nearest Node Heuristic

Για τη δημιουργία της αρχικής λύσης αλλά και κατά την εκτέλεση της τοπικής αναζήτησης χρησιμοποιείται ο αλγόριθμος του πλησιέστερου γείτονα. Χαρακτηριστικό του αλγορίθμου είναι η ταχύτητα στην οποία δίνει αποτελέσματα ωστόσο δεν είναι ιδιαίτερα αποτελεσματικός. Ο λόγος που χρησιμοποιείται είναι το γεγονός ότι δίνει μια τάξη μεγέθους για τις τιμές των λύσεων και σύμφωνα με αυτή την τιμή αρχικοποιείται η φερομόνη στο γράφημα. Επίσης, χρησιμοποιείται παράλληλα με την εκτέλεση του αλγορίθμου 1-1 exchange έτσι ώστε να ανασυντάξει το εκάστοτε δρομολόγιο. Η ανασύνταξη αυτή κατά πάσα πιθανότητα δεν είναι η

βέλτιστη, όμως είναι πολύ πιο σύντομη η δημιουργία της σε σύγκριση με τον 2-opt. Την περεταίρω βελτιστοποίηση θα την κάνουν τα μυρμήγκια των επόμενων επαναλήψεων. Η διαδικασία που ακολουθεί ο αλγόριθμος είναι εξαιρετικά απλή. Ξεκινώντας από την αποθήκη επιλέγει να επισκεφτεί τον κόμβο ο οποίος βρίσκεται πλησιέστερα σε αυτόν. Έπειτα επαναλαμβάνει την ίδια διαδικασία επιλογής έχοντας ως αφετηρία τον κόμβο που βρίσκεται έως ότου το απόθεμα δεν επαρκεί για την επόμενο κόμβο, οπότε και γυρνάει στην αποθήκη. Η λύση ολοκληρώνεται όταν όλοι οι κόμβοι συμπεριλαμβάνονται στη λύση. Για την εφαρμογή του κατά την τοπική αναζήτηση μέσα στο ίδιο δρομολόγιο προφανώς η διαδικασία λαμβάνει χώρα για μια διαδρομή από και προς την αποθήκη αφού οι περιορισμοί χωρητικότητας είναι εξασφαλισμένοι.

1-1 Exchange

Ο αλγόριθμος 1-1 Exchange ανταλλάσει 2 κόμβους που βρίσκονται σε διαφορετικές διαδρομές. Η επιλογή των διαδρομών γίνεται τυχαία ή σειριακά ή με κάποιο κριτήριο που σχετίζεται με το πρόβλημα για έναν αριθμό επαναλήψεων και έπειτα ελέγχονται όλες οι εφικτές ανταλλαγές μεταξύ αυτών και επιστρέφεται η καλύτερη λύση. Ο αλγόριθμος αυτός αφορά την δια-δρομολογιακή (inter-route) βελτιστοποίηση που σα στόχο έχει την απομάκρυνση ενός "κακού" κόμβου από μια διαδρομή και τη ανταλλαγή του με έναν "κακό" κόμβο μιας άλλης. Υπάρχουν άλλοι αλγόριθμοι αυτής της κατηγορίας όπως 2-2 exchange και οι αλγόριθμοι relocate όπου αντί για ανταλλαγή κόμβων έχουμε μετακίνηση από μια διαδρομή σε μια άλλη.



Στην επόμενη ενότητα εξηγούνται αναλυτικά όλες οι εφαρμογές του 1-1 exchange που χρησιμοποιήθηκαν για την επίλυση του προβλήματος. Παρουσιάζονται οι τρόποι επιλογής των διαδρομών που δοκιμάστηκαν συνοδευόμενοι από τα αποτελέσματα τους.

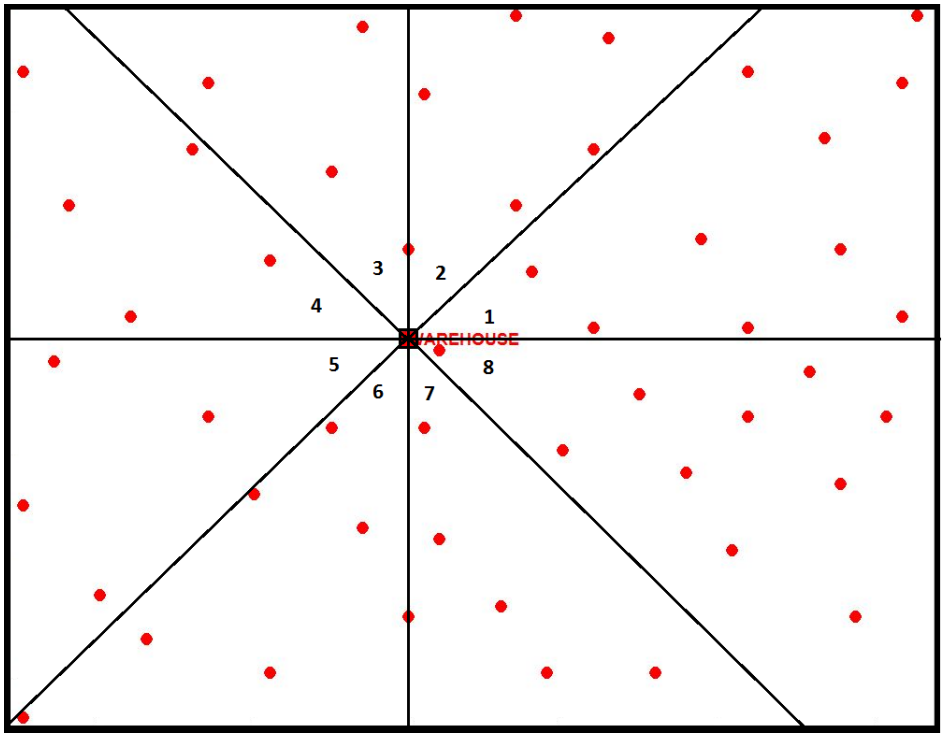
Rearrange

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου πολλές από τις λύσεις που θα δημιουργηθούν περιλαμβάνουν περισσότερες διαδρομές από τις ελάχιστες απαραίτητες. Για το λόγο αυτό δημιουργήθηκε μια μεθοδολογία ώστε σε σύντομο χρόνο να ανασυνταχτούν τα δρομολόγια ώστε να χωρέσουν όλοι οι κόμβοι στον ελάχιστο απαιτούμενο αριθμό διαδρομών. Αν παρόλα αυτά δεν καταφέρουν να χωρέσουν η λύση απορρίπτεται και το μυρμήγκι ξεκινάει από την αρχή. Η διαδικασία που ακολουθείται είναι η εξής:

```
-Χώρισε τη λύση σε δρομολόγια
-Υπολόγισε το διαθέσιμο απόθεμα κάθε δρομολογίου
ΟΣΟ τα δρομολόγια είναι περισσότερα από τα ελάχιστα απαιτούμενα
        -Κατέταξε τα δρομολόγια σύμφωνα με το διαθέσιμο απόθεμα(Min->Max)
        -Κατέταξε τους κόμβους που βρίσκονται στην επιπλέον διαδρομή συμφωνα με τη ζήτηση(Max->Min)
        -Ξεκινώντας απο το δρομολόγιο 1 και κόμβο 1
        ΚΑΝΕ
                ΑΝ χωράει ο κόμβος στο δρομολόγιο
                        -Βάλτον στο δρομολόγιο
                        -Αφαίρεσε τη ζήτηση από το διαθέσιμο απόθεμα
                        -Επόμενος κόμβος
                ΑΛΛΙΩΣ
                        -Επόμενο δρομολόγιο
                ΤΕΛΟΣ_ΑΝ
                ΑΝ τοποθετήθηκαν όλοι οι επιπλεόν κόμβοι
                        -Επέστρεψε: Δυνατό
                        -Σταματα
                ΤΕΛΟΣ_ΑΝ
                ΑΝ το δρομολόγιο είναι μεγαλύτερο από τον ελάχιστο απαιτούμενο αριθμό
                        -Επέστρεψε: Αδύνατο
                        -Σταμάτα
                ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΚΑΝΕ
ΤΕΛΟΣ_ΟΣΟ
```

Pseudo clustering

Σε μια από τις παραλλαγές που δημιουργήθηκαν αναζητήθηκε ένα κριτήριο επιλογής των δρομολογίων στα οποία εφαρμόζεται ο αλγόριθμος 1-1 exchange. Για το λόγο αυτό οι κόμβοι του γραφήματος ομαδοποιήθηκαν ως εξής: Μεταφέραμε τις συντεταγμένες όλως των κόμβων στο γράφημα καταλλήλως έτσι ώστε η αποθήκη να βρίσκεται στο σημείο με συντεταγμένες (0,0). Έπειτα, γνωρίζοντας τις συντεταγμένες των κόμβων υπολογίσαμε την εφαπτομένη και στη συνέχεια τη γωνία που σχηματίζει στο νοητό κύκλο με κέντρο την αποθήκη. Τέλος, χωρίσαμε το γράφημα σε ομάδες ανά διαστήματα $45^{o}$, δηλαδή 8 ομάδες που περιλαμβάνουν τους κόμβους που βρίσκονται μέσα σε αυτές τις γωνίες. Παρακάτω φαίνεται γραφικά πως χωρίστηκε το γράφημα του προβλήματος par1 σε ομάδες.




Στην επόμενη ενότητα γίνεται η παρουσίαση των διαφόρων παραλλαγών που δοκιμάστηκαν για την επίλυση των προβλημάτων, όπου γίνεται χρήση των αλγόριθμων και των μεθόδων που περιγράφηκαν παραπάνω. Η υλοποίηση των αλγορίθμων έγινε στη γλώσσα προγραμματισμού C++11 στο IDE Microsoft Visual Studio 2013.

## 4.2 Εφαρμογές

Τα προβλήματα τα οποία τρέξαμε είναι τα par1(50), par2(15), par3(100), par4(150), par11(121), par12(101) καθώς και προβλήματα από το A-n32κ5 μέχρι και A-n80κ10. Τα δεδομένα αυτά αρχικά δημιουργήθηκαν για το πρόβλημα απλό VRP, ωστόσο στη βιβλιογραφία χρησιμοποιούνται και για το CCVRP. Για τα προβλήματα par και 3 εκ των προβλημάτων An#k# χρησιμοποιήθηκαν και οι 5 εκδοχές του αλγορίθμου ενώ για τα υπόλοιπα μόνο οι δύο αποτελεσματικότερες. Στη συνέχεια παρουσιάζονται αναλυτικά οι παραλλαγές του κώδικα μαζί με τα αποτελέσματα τους. Η σύγκριση των αποτελεσμάτων των προβλημάτων par και An#k# γίνεται με τα αποτελέσματα από τις δημοσιεύσεις [5] και [8] αντίστοιχα. Εξαίρεση το πρόβλημα An32k5 που η σύγκριση γίνεται με το αποτέλεσμα από την [6].

AS-Ranked simple

Η πρώτη εκδοχή είναι η απλούστερη καθώς περιλαμβάνει την εφαρμογή του Ant System Ranked όπως περιγράφηκε στην προηγούμενη παράγραφο μαζί με τη μέθοδο Rearrange. Ο λόγος που γίνεται η ανασύνταξη αυτή είναι διότι σε μερικά προβλήματα παρουσιάζονται πολλές λύσεις που δεν είναι εφικτές και επομένως δεν μπορούσαμε να τις απορρίπτουμε όλες. Έτσι, οι επιλογές ήταν να τις δεχόμαστε με κάποιο penalty function ή να προσπαθήσουμε να τις μετατρέψουμε σε εφικτές. Ο κύριος λόγος που επιλέχτηκε η προσπάθεια μετατροπής σε εφικτή είναι διότι έτσι αυτομάτως δημιουργείται μια εφικτή λύση η οποία συνεισφέρει στην αναζήτηση των μυρμηγκιών με το πραγματικό της κόστος, δίχως την εισαγωγή μιας υποκειμενικής ποινής η οποία με τη σειρά της θα απαιτούσε ρύθμιση ανάλογα με το εύρος τιμών των λύσεων του εκάστοτε προβλήματος. Για παράδειγμα όταν στις λύσεις των μυρμηγκιών υπάρχουν εφικτές και ανέφικτες λύσεις, υπάρχει κίνδυνος να επιλέγονται οι ανέφικτες λύσεις έναντι των εφικτών επειδή η ποινή δεν είναι καλά ρυθμισμένη ή/και η εφικτές λύσεις είναι πολύ κακές. Για να λυθεί ένα τέτοιο ενδεχόμενο, θα απαιτούσε τη χρήση μιας διαδικασίας που θα δίνει προτεραιότητα στις εφικτές λύσεις ανεξαρτήτου κόστους.

Επιπλέων, πέραν της ανάγκης ρύθμισης του συντελεστή ποινής ένα άλλο μειονέκτημα της συνάρτησης ποινής είναι ότι σε ακραίες περιπτώσεις όπου όλες οι λύσεις είναι ανέφικτές, ο αλγόριθμος αρχίζει να συγκλίνει και να ψάχνει τη βέλτιστη μεταξύ των ανέφικτων λύσεων, δηλαδή ο αλγόριθμος ουσιαστικά αποτυγχάνει. Στα πλεονεκτήματα της συνάρτησης ποινής είναι ο ελάχιστος υπολογιστικός χρόνος που απαιτείται έναντι της προσπάθειας ανασύνταξης της λύσης. Έτσι, μεταξύ μιας στρατηγικής "αποτρεπτικής" προτιμήθηκε μια στρατηγική "εποικοδομητική", χωρίς να σημαίνει ότι για όλα τα προβλήματα θα υπερτερεί η μία της άλλης.

AS-Ranked + Fuzzy Alpha

Παρατηρώντας την αλληλουχία των καλύτερων λύσεων κατά την εκτέλεση του AS-Ranked παρατηρήθηκε ότι πολλές φορές ο αλγόριθμος κολλούσε σε τοπικά βέλτιστα. Ειδικότερα, όταν η διαφορά με την προηγούμενη καλύτερη λύση ήταν μεγάλη η ποσότητα φερομόνης μεταξύ των άλλων λύσεων και την νέας βέλτιστης ήταν αρκετά μεγάλη ώστε να περιοριστεί πολύ το πλάτος της αναζήτησης. Αποτέλεσμα αυτού ήταν τα μυρμήγκια να περιορίζονται στην αναζήτηση λύσεων και να καταλήγουν να εγκλωβίζονται σε μια συγκεκριμένη περιοχή.

Για να αποφθεχθεί αυτή η στασιμότητα και να απεγκλωβιστεί ο αλγόριθμος από το τοπικό βέλτιστο, προχωρήσαμε στην μετατροπή της παραμέτρου Alpha από στατική σε δυναμική και προσαρμοζόμενη ανάλογα με της ανάγκες του αλγορίθμου. Επιπλέον, θέλαμε η ρύθμιση αυτή να είναι ανεξάρτητη του μεγέθους του προβλήματος και του εύρους τιμών των λύσεων. Έτσι, ως μεταβλητή εισαγωγής χρησιμοποιήθηκε μόνο ο αριθμός των επαναλήψεων όπου το καλύτερο μυρμήγκι της

επανάληψης είναι ίσο με το καλύτερο έως τώρα μυρμήγκι. Ο αριθμός αυτός μηδενίζεται όταν βρεθεί μια νέα βέλτιστη λύση.

Μεταβάλλοντας την παράμετρο $\boldsymbol{\alpha}$ στο διάστημα [0,1], οι τιμές της φερομόνης κατά τον υπολογισμό των πιθανοτήτων μετάβασης γίνονται ποιο δυσδιάκριτες μεταξύ τους ή αλλιώς συμπυκνώνονται σε τιμές πιο κοντά μεταξύ τους δίνοντας έτσι μεγαλύτερες πιθανότητες στα μυρμήγκια να επιλέξουν κάποια διαδρομή που δεν έχει επιλεγεί μέχρι εκείνη τη στιγμή. Όντας κοντά μεταξύ τους, οι τιμές, αποφεύγεται το φαινόμενο κατά το οποίο μια διαδρομή επιλέγεται από τη συντριπτική πλειοψηφία των μυρμηγκιών λόγο της διαφοράς στα επίπεδα φερομόνης από τη δεύτερη, τρίτη κλπ καλύτερη διαδρομή, εγκλωβίζοντας τα μυρμήγκια σε ένα τοπικό ελάχιστο.

AS-Ranked + random Local Search+ Fuzzy Alpha

Μετά τη βελτίωση των αποτελεσμάτων από την προσθήκη του μεταβαλλόμενου Alpha, όπου αποδείχθηκε ότι η προσθήκη αυτή έχει θετικό αντίκτυπο στον αλγόριθμο, στόχος έγινε η περαιτέρω βελτίωση των αποτελεσμάτων ώστε να πλησιάσουν τα βέλτιστα της βιβλιογραφίας. Για το λόγο αυτό εισήχθη μια διαδικασία τοπικής αναζήτησης. Αυτή η προσπάθεια τοπικής βελτιστοποίησης της λύσης περιλαμβάνει τη μέθοδο 1-1 exchange με τυχαία επιλογή των δρομολογίων που θα ανταλλάξουν δρομολόγια. Για αυτά τα 2 δρομολόγια που επελέγησαν δοκιμάζονται όλες οι ανταλλαγές κόμβων και αυτές οι οποίες είναι εφικτές επιδέχονται μια ενδο-δρομολογιακή (intra-route) βελτιστοποίηση με τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Node Heuristic). Αν το άθροισμα του κόστους των νέων δρομολογίων είναι μικρότερο από το αρχικό κρατάμε το νέο ως λύση.

Ο λόγος που για χρησιμοποιήθηκε ο αλγόριθμος του πλησιέστερου γείτονα έναντι κάποιου άλλου, όπως ο 2-opt ή ο 3-opt, είναι η τεράστια διαφορά στον χρόνο που απαιτούν για την εκτέλεση τους. Πέραν αυτού, σε προβλήματα όπως το απλό VRP ή το TSP για να επιλεγεί η όχι μια 2-opt κίνηση ως συμφέρουσα αρκεί η σύγκριση του κόστους των δυο κινήσεων. Για το Cumulative CVRP όμως απαιτείται ο υπολογισμός του κόστους όλου του δρομολογίου πριν και μετά την αλλαγή αυτή, πράγμα που καθιστά τον 2opt ακόμα πιο αργό. Επομένως, θεωρήθηκε συμφέρων μια αρκετά ικανοποιητική λύση πάνω στην οποία τα επόμενα μυρμήγκια θα μπορέσουν να ψάξουν για μονοπάτια ακόμα πιο οικονομικά.

AS-Ranked + extensive Local Search+ Fuzzy Alpha

Παρόλη την προσθήκη της τοπικής αναζήτησης παρατηρήσαμε ότι ορισμένες διαδρομές εξακολουθούν να συμπεριλαμβάνουν κόμβους αρκετά απομακρυσμένους από τους υπόλοιπους στερώντας μας έτσι μια καλύτερη λύση. Σε αυτή την τροποποίηση επεκτείναμε την τοπική αναζήτηση προκειμένου περισσότερες διαδρομές να επιδέχονται την το 1-1 exchange. Έτσι, αντί για την επιλογή 2 τυχαίων διαδρομών προς εξέταση, ο αλγόριθμος εξετάζει όλους του δυνατούς συνδυασμούς διαδρομών έως ότου δεν υπάρχει περεταίρω βελτίωση μετά από έναν αριθμό συνδυασμού διαδρομών.

Η μετατροπή αυτή όπως είναι φυσικό κάνει τον αλγόριθμο αρκετά πιο αργό, ωστόσο βελτιώνει τα αποτελέσματα σε όλα τα προβλήματα εκτός από το par1 όπου και προηγούμενες εκδοχές συνέκλιναν στο ολικό βέλτιστο της βιογραφίας.

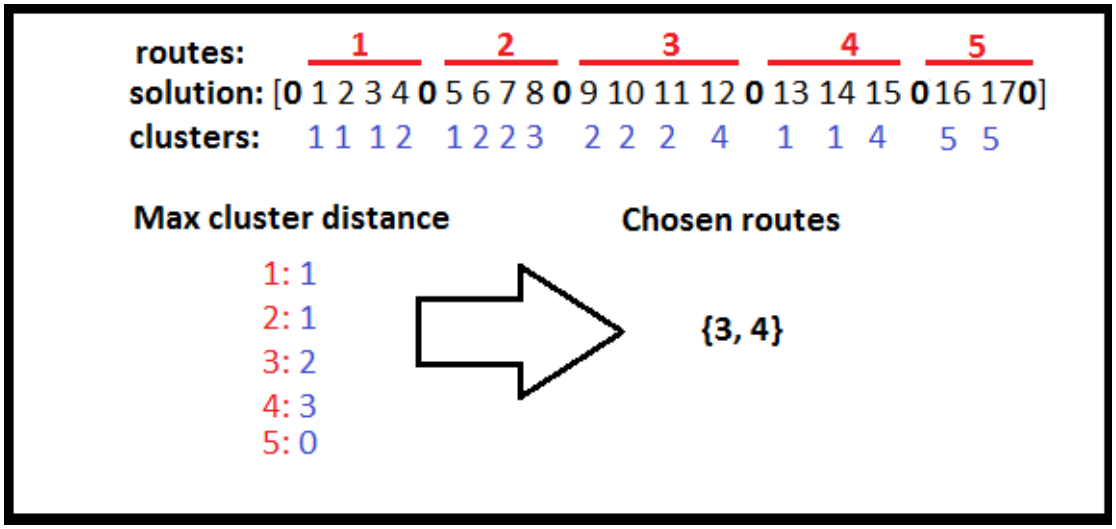
AS-Ranked + pseudo cluster Local Search+ Fuzzy Alpha

Το πρόβλημα με την προηγούμενη παραλλαγή της τοπική αναζήτησης ήταν ο χρόνος τον οποίο χρειαζόταν για να εκτελεστεί. Για το λόγο αυτό θεωρήθηκε ότι η επιλογή των διαδρομών προς αντιμετάθεση κόμβων είναι καλύτερα να μην γίνεται τυχαία αλλά να επιλέγονται βάσει ενός κριτηρίου που θα αποτελεί ένδειξη για το πόσο πιθανό είναι η εκάστοτε διαδρομή να χρειάζεται αντιμετάθεση για να βελτιωθεί η λύση.

Προκειμένου να βρούμε αυτό το κριτήριο επιλογής, αρχικά εισαγάγαμε στο πρόβλημα την ομαδοποίηση των κόμβων (pseudo clusters). Έτσι κατά την εισαγωγή των δεδομένων του προβλήματος πέραν του υπολογισμού των αποστάσεων των κόμβων, κάθε κόμβος καταχωρείται σε μια ομάδα. Η διαδικασία με την οποία κάθε κόμβος καταχωρείται σε μια ομάδα περιγράφηκε στην προηγούμενη ενότητα.

Η χρήση της πληροφορίας της ομάδας στην οποία ανήκει ο κάθε κόμβος, βοηθάει στην εύρεση των δυο δρομολογίων με τις μεγαλύτερες πιθανότητες βελτίωσης (most promising) ως εξής: Οι λύσεις χωρίζονται σε δρομολόγια, τα οποία αντιστοιχούν σε ένα διάνυσμα ομάδων. Για κάθε δρομολόγιο υπολογίζεται η μέγιστη τιμή διαφοράς συνεχόμενων ομάδων που περιέχουν. Τα δυο δρομολόγια με το μεγαλύτερη μέγιστη διαφορά επιλέγονται για 1-1 exchange. Ουσιαστικά, θεωρούμε ότι μια διαδρομή που περιέχει κάποιο κόμβο απομακρυσμένης ομάδας ως προς του υπόλοιπους, έχει περισσότερες πιθανότητες να χρήζει τοπικής βελτιστοποίησης από μια διαδρομή που όλοι οι κόμβοι βρίσκονται σε κοντινές ομάδες.

Δίνεται παράδειγμα της διαδικασίας επιλογής διαδρομών προς τοπική βελτιστοποίηση:



Παρατηρούμε ότι επιλέγονται οι διαδρομές 3 και 4 διότι σε αυτές οι μέγιστη απόσταση ομάδων είναι 2 και 3 αντίστοιχα. Αυτή η απόσταση προκύπτει από την τη διαφορά ομάδας μεταξύ των κόμβων 11- 12 (διαδρομή 3) και 14-15 (διαδρομή 4) όπου είναι 2-4 και 1-4 αντίστοιχα.

Παρόλα αυτά επειδή ναι μεν αυτή η μέθοδος επιλογής είναι συντομότερη αλλά όχι απαραίτητα αποτελεσματικότερη συνδυάσαμε τις 2 μεθόδους επιλογής των διαδρομών προς ανταλλαγή κόμβων σε ποσοστά 98% με χρήση των pseudo cluster και με πιθανότητα 2% σειριακή επιλογή των συνδυασμών.

## 4.3 Αποτελέσματα

Για όλες τις παραλλαγές του αλγορίθμου που δοκιμάστηκαν οι αρχικές παράμετροι ρυθμίστηκαν όπως προτείνεται στο [1], δηλαδή α=1, β=4, Ants=NumberOfNodes. Λόγο της προσθήκης του Fuzzy Alpha η τιμή του α σε αυτές τις εφαρμογές αλλάζει κατά την εκτέλεση του αλγορίθμου. Για τη σύγκριση των αποτελεσμάτων τόσο μεταξύ των αλγορίθμων όσο και ανάμεσα στα αποτελέσματα τους και τα βέλτιστα της βιογραφίας (BKS) χρησιμοποιήθηκε η σχετική ποσοστιαία απόκλιση (RPD) που δίνεται από τον τύπο:

$$\mathrm{RPD} = (\mathrm{F} - \mathrm{F}_{BKS}) / \mathrm{F}_{BKS} \times 100$$

Μια έννοια που θα χρησιμοποιήσουμε στην ανάλυση των αποτελεσμάτων είναι αυτή της σφικτότητας (tightness). Είναι μια ένδειξη της δυσκολίας ευρέσεως εφικτών λύσεων και μας βοηθάει στην καλύτερη κατανόηση της συμπεριφοράς του αλγορίθμου. Ο υπολογισμός της γίνεται από τον ακόλουθο τύπο:

$$tightness = \frac{TotalDemand}{Capacity \times MinFleet}$$

Το μέτρο αυτό δεν είναι απόλυτο αλλά δείχνει σε μεγάλο βαθμό πόσο δύσκολο ή εύκολο είναι ένα πρόβλημα. Δεν λαμβάνει υπόψη τη διακύμανση των τιμών των επιμέρους ζητήσεων των κόμβων που σαφώς και παίζουν ρόλο στην ευκολία σχηματισμού λύσεων. Για παράδειγμα, όταν υπάρχουν πολλοί κόμβοι με μικρή ζήτηση είναι ευκολότερη η εύρεση εφικτής λύσης σε σύγκριση με ένα πρόβλημα λιγότερων κόμβων αλλά με μεγαλύτερες τιμές ζητήσεων. Στην πρώτη περίπτωση υπάρχουν πολλοί διαθέσιμοι κόμβοι που μπορούν να αντιμετατεθούν σχηματίζοντας νέες εφικτές λύσεις, ενώ στη δεύτερη, σαφώς λιγότερες επιλογές αν και στο δείκτη σφικτότητας μπορούν να έχουν την ίδια τιμή. Στον πίνακα που ακολουθεί φαίνονται τα ποσοστά σφικτοτητας για τα προβλήματα που επιλύθηκαν:

| | Fleet | Capacity | Tot. Demand | %tightness |
|---|---|---|---|---|
| **par1** | 5 | 160 | 777 | 97.125 |
| **par2** | 10 | 140 | 1364 | 97.4285714 |
| **par3** | 8 | 200 | 1458 | 91.125 |
| **par4** | 12 | 200 | 2235 | 93.125 |
| **par11** | 7 | 200 | 1375 | 98.2142857 |
| **par12** | 10 | 200 | 1810 | 90.5 |
| **n32k5** | 5 | 100 | 410 | 82 |
| **n33k5** | 5 | 100 | 446 | 89.2 |
| **n69k9** | 9 | 100 | 845 | 93.8888889 |
| **n37k5** | 5 | 100 | 407 | 81.4 |
| **n39k6** | 6 | 100 | 526 | 87.6666667 |
| **n48k7** | 7 | 100 | 626 | 89.4285714 |
| **n53k7** | 7 | 100 | 664 | 94.8571429 |
| **n62k8** | 8 | 100 | 733 | 91.625 |
| **n63k9** | 9 | 100 | 873 | 97 |
| **n80k10** | 10 | 100 | 942 | 94.2 |

Σύγκριση AS-Ranked Simple και AS-Ranked + Fuzzy Alpha

Προκειμένου να διαπιστωθεί αν η προσθήκη της ρυθμιζόμενης τιμής της παραμέτρου Alpha εξυπηρετούσε τον αλγόριθμο βελτιώνοντας τα αποτελέσματα του, τρέξαμε τα προβλήματα par και ορισμένα εκ των An#k#. Στον παρακάτω πίνακα φαίνονται τα βέλτιστα αποτελέσματα για τους δύο αλγορίθμους μετά από 5 τρεξίματα καθώς και η μέση βελτίωση των αποτελεσμάτων με τη χρήση του Fuzzy Alpha.

| | **ASR** | | **ASR+alpha** | | **BKS** | ASR - ASR+alpha |
|---|---|---|---|---|---|---|
| | *Best* | *RPD* | *Best* | *RPD* | | RPD |
| **par1** | 2238.02 | 0.343892 | 2230.35 | 0 | **2230.35** | 0.343892214 |
| **par2** | 2553.8 | 6.780731 | 2478.8 | 3.644795 | **2391.63** | 3.025657576 |
| **par3** | 4301.67 | 6.334324 | 4210.88 | 4.090057 | **4045.42** | 2.156081389 |
| **par4** | 5406.7 | 8.404578 | 5364.6 | 7.560471 | **4987.52** | 0.784774261 |
| **par11** | 8408.11 | 14.92973 | 8181.72 | 11.83523 | **7315.87** | 2.767022093 |
| **par12** | 3835.7 | 7.777078 | 3835.7 | 8.591089 | **3558.92** | 0 |
| **n32k5** | 2278.83 | 3.961223 | 2270.05 | 3.560675 | **2192** | 0.386775622 |
| **n33k5** | 1744.98 | -1.47868 | 1739.81 | -1.77058 | 1771.17 | 0.297158885 |
| **n69k9** | 3722.47 | -1.5644 | 3689.19 | -2.44445 | 3781.63 | 0.90209504 |

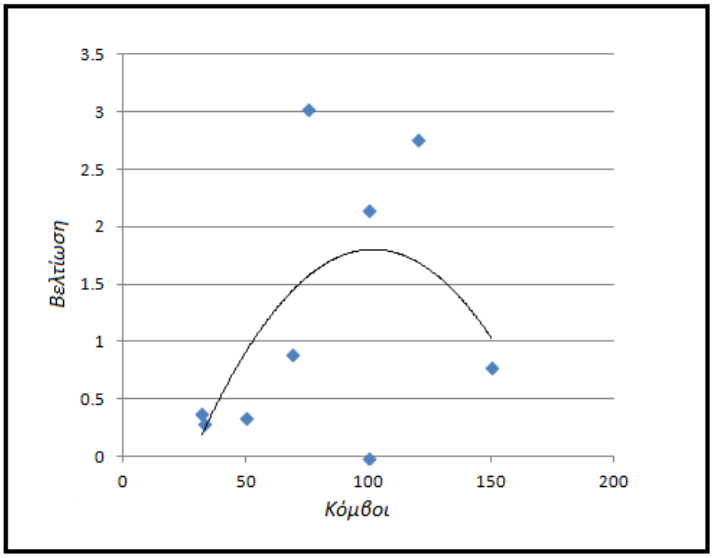
Από τον πίνακα αποτελεσμάτων εύκολα συμπεραίνεται ότι η χρήση την μεταβλητής παραμέτρου Alpha βοήθησε τον αλγόριθμο να ξεκολλήσει από κάποια

τοπικά βέλτιστα και να αναζητήσει νέες λύσεις. Σε όλα τα προβλήματα υπάρχει βελτίωση του αποτελέσματος με εξαίρεση το par12 όπου δεν φαίνεται να το επηρεάζει θετικά αλλά ούτε αρνητικά. Η μέση βελτίωση που επιδέχονται τα αποτελέσματα από την προσθήκη του Fuzzy Alpha είναι 1,184%.

Από την ποσοστιαία βελτίωση ανά πρόβλημα είναι ασφαλές να πούμε ότι το μέγεθος του προβλήματος δεν παίζει κάποιο ιδιαίτερο ρόλο στην αποτελεσματικότητα του Fuzzy Alpha καθότι δεν παρατηρείται κάποια τάση αύξησης ή μείωσης της ανάλογα με το πλήθος των κόμβων στο πρόβλημα. Επομένως, η διασπορά αυτή των τιμών οφείλεται περισσότερο στη τοπολογία του εκάστοτε προβλήματος και του κατά πόσο είναι εύκολο ή δύσκολο να παγιδευτεί ο αλγόριθμος σε κάποιο τοπικό βέλτιστο στο συγκεκριμένο γράφημα. Όσο πιο εύκολο είναι να συγκλίνει σε κάποιο τοπικό βέλτιστο τόσο πιο χρήσιμη η μεταβολή της παραμέτρου.

Για παράδειγμα το πρόβλημα par2 έχει ως χαρακτηριστικό τη δυσκολία εύρεσης εφικτών λύσεων, διότι ο περιορισμός χωρητικότητας των οχημάτων είναι πολύ σφικτός. Αποτέλεσμα αυτού είναι να μη υπάρχει μεγάλη ποικιλία εφικτών λύσεων κατά τη αναζήτηση λύσεων από τα μυρμήγκια, καθιστώντας εύκολη τη σύγκλιση σε ένα τοπικό βέλτιστο. Στο συγκεκριμένο πρόβλημα παρατηρείται και η μεγαλύτερη βελτίωση του αποτελέσματος.

Αντίθετα, στο πρόβλημα par12 το οποίο έχει 25 περισσότερους κόμβους από το par2 των 75 κόμβων η βελτίωση της λύσης είναι μηδενική. Το πρόβλημα Par11 έχει 120 κόμβους, δηλαδή 20 περισσότερους από το par12, εμφανίζει βελτίωση της τάξεως του 2,76%. Είναι μια αρκετά σαφής ένδειξη ότι η βελτίωση των αποτελεσμάτων είναι ανεξάρτητη του αριθμού των κόμβων του προβλήματος. Στο παρακάτω γράφημα φαίνεται η γραμμή τάσης η οποία δεν είναι μονότονη και ως εκ τούτου επιβεβαιώνει τις παρατηρήσεις που έγιναν προηγουμένως.

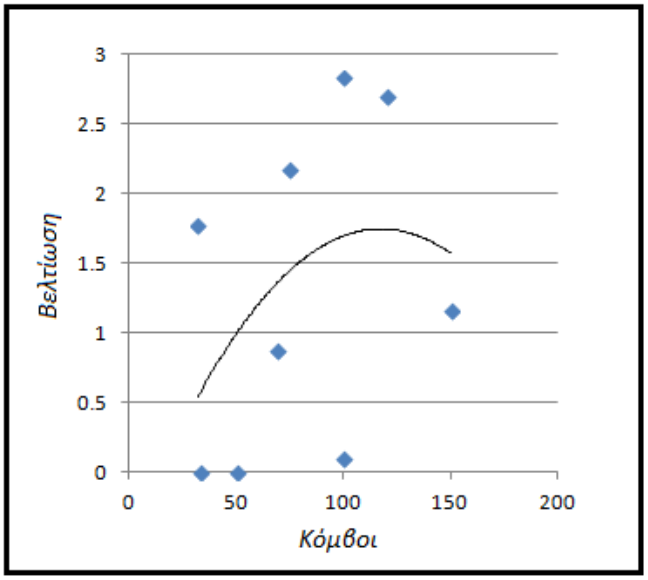


Σύγκριση AS-Ranked + Fuzzy Alpha με Random LS και χωρίς LS

Όπως και για την παραπάνω σύγκριση και σε αυτή τρέξαμε τα προβλήματα 5 φορές κρατώντας τα καλύτερα αποτελέσματα που έδωσαν προκειμένου να μελετήσουμε την αποτελεσματικότητα της προσθήκης την τοπικής αναζήτησης τύπου Random όπως περιγράφηκε παραπάνω. Ακολουθεί ο συνοπτικός πίνακας αποτελεσμάτων που περιέχει τις τιμές των αποτελεσμάτων, την απόκλιση από το βέλτιστο καθώς και την ποσοστιαία βελτίωση που προσφέρει η τοπική αναζήτηση σε σχέση με την εκδοχή του αλγορίθμου χωρίς αυτή.

| | **ASR+alpha** | | **ASR+alpha+randLS** | | **BKS** | ASR w/o LS - ASR w/LS |
|---|---|---|---|---|---|---|
| | *Best* | *RPD* | *Best* | *RPD* | | RPD |
| **par1** | 2230.35 | 0 | 2230.35 | 0 | **2230.35** | 0 |
| **par2** | 2478.8 | 3.644795 | 2426.02 | 1.437931 | **2391.63** | 2.175579756 |
| **par3** | 4210.88 | 4.090057 | 4206.62 | 3.984753 | **4045.42** | 0.101268952 |
| **par4** | 5364.6 | 7.560471 | 5302.77 | 6.320777 | **4987.52** | 1.165994377 |
| **par11** | 8181.72 | 11.83523 | 7966.35 | 8.891355 | **7315.87** | 2.703496583 |
| **par12** | 3835.7 | 7.777078 | 3730.13 | 4.810729 | **3558.92** | 2.830196267 |
| **n32k5** | 2270.05 | 3.560675 | 2230.6 | 1.760949 | **2192** | 1.768582444 |
| **n33k5** | 1739.81 | -1.77058 | 1739.81 | -1.77058 | 1771.17 | 0 |
| **n69k9** | 3689.19 | -2.44445 | 3657.33 | -3.28694 | 3781.63 | 0.871127298 |

Παρατηρώντας τα αποτελέσματα της σύγκρισης, συμπεραίνεται ότι η χρήση της τοπικής αναζήτησης βοήθησε στην εξαγωγή καλύτερων αποτελεσμάτων, βελτιώνοντας τα μεσοσταθμικά κατά 1,29%. Και σε αυτές τις βελτιώσεις δεν υπάρχει ξεκάθαρη τάση που να υποδηλώνει ότι το μέγεθος του προβλήματος παίζει σημαντικό ρόλο στην ποσοστιαία βελτίωση που παρουσιάζεται. Επομένως, παρατηρώντας και το παρακάτω διάγραμμα τάσης μπορούμε να συμπεράνουμε ότι η μορφολογία του γραφήματος παίζει το σημαντικότερο ρόλο.



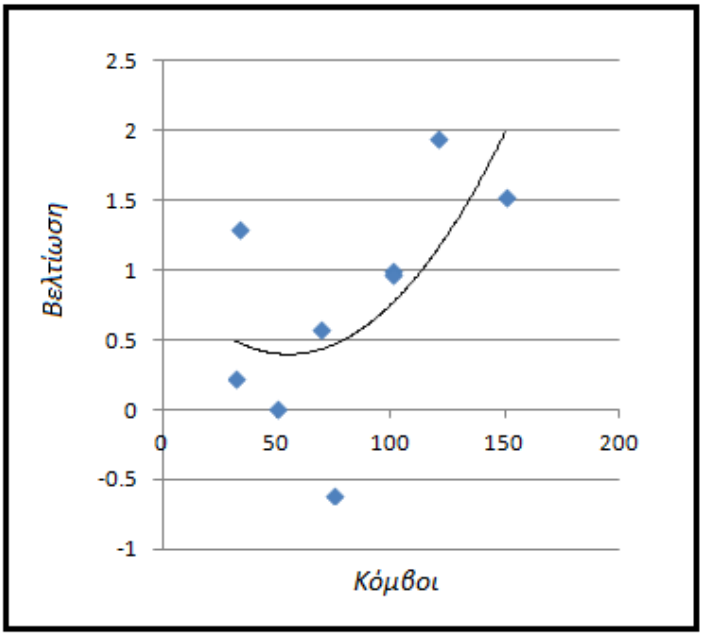
Σύγκριση AS-Ranked + Fuzzy Alpha με Random LS και extensive LS

Η επέκταση της τοπικής αναζήτησης είχε ως αποτέλεσμα ο αλγόριθμος να γίνει αρκετά πιο αργός ωστόσο βελτίωσε περεταίρω τα αποτελέσματα στις περισσότερες των περιπτώσεων. Στον παρακάτω πίνακα φαίνεται η βελτίωση που είχε σε κάθε πρόβλημα ξεχωριστά ενώ η μέση βελτίωση ήταν 0,76%.

| | **ASR+alpha+randLS** | | **ASR+alpha+extLS** | | **BKS** | ASR randLS - ASR extLS |
|---|---|---|---|---|---|---|
| | *Best* | *RPD* | *Best* | *RPD* | | RPD |
| **par1** | 2230.35 | 0 | 2230.35 | 0 | **2230.35** | 0 |
| **par2** | 2426.02 | 1.437931 | 2441 | 2.064283 | **2391.63** | -0.613682917 |
| **par3** | 4206.62 | 3.984753 | 4165.1 | 2.958407 | **4045.42** | 0.996854817 |
| **par4** | 5302.77 | 6.320777 | 5222.99 | 4.721184 | **4987.52** | 1.527477556 |
| **par11** | 7966.35 | 8.891355 | 7815.05 | 6.823249 | **7315.87** | 1.936008087 |
| **par12** | 3730.13 | 4.810729 | 3694.32 | 3.804525 | **3558.92** | 0.969325884 |
| **n32k5** | 2230.6 | 1.760949 | 2225.59 | 1.532391 | **2192** | 0.225108848 |
| **n33k5** | 1739.81 | -1.77058 | 1717.71 | -3.01834 | 1771.17 | 1.28659669 |
| **n69k9** | 3657.33 | -3.28694 | 3636.66 | -3.83353 | 3781.63 | 0.568378677 |

Στο πρόβλημα par2 όχι μόνο δεν υπήρξε βελτίωση αλλά αντίθετα ο αλγόριθμος παγιδεύτηκε σε κάποιο τοπικό ελάχιστο από το οποίο δεν κατάφερε να ξεφύγει. Όπως αναφέρθηκε και σε προηγούμενη ανάλυση το πρόβλημα αυτό χαρακτηρίζεται από τη δυσκολία εύρεσης εφικτών λύσεων λόγο του περιορισμού χωρητικότητας. Για το λόγο αυτό είναι πιθανό ο αλγόριθμος κατά την εκτενή τοπική αναζήτηση να καταλήγει στο ίδιο τοπικό βέλτιστο από τις πρώτες κιόλας επαναλήψεις μη επιτρέποντας στα μυρμήγκια να εξερευνήσουν όλο το πλάτος του χώρου των λύσεων αποτελεσματικά. Αυτό έχει ως αποτέλεσμα την πρόωρη σύγκλιση του αλγορίθμου σε μια μη ικανοποιητική λύση.

Θέλοντας να μελετήσουμε το συσχετισμό ανάμεσα στο μέγεθος του προβλήματος και την βελτίωση του αποτελέσματος από την επέκταση της τοπικής αναζήτησης ανατρέχουμε ξανά στο διάγραμμα της τάσης που βρίσκεται παρακάτω. Αν και τα δεδομένα δεν είναι πάρα πολλά, μπορούμε να εξάγουμε το εξής συμπέρασμα: Στα μικρά προβλήματα, μικρότερα των ~75 κόμβων το μέγεθος του προβλήματος δεν παίζει τόσο σημαντικό ρόλο όσο η ευκολία σχηματισμού εφικτών λύσεων. Εκεί που η επέκταση της τοπικής αναζήτησης φαίνεται να σχετίζεται με το μέγεθος του προβλήματος είναι στα μεγαλύτερα προβλήματα άνω των ~75 κόμβων όπου φαίνεται και στο διάγραμμα μια ξεκάθαρη αύξουσα τάση.




Σύγκριση AS-Ranked + Fuzzy Alpha με Random LS και pseudo cluster LS

Θέλοντας να μειώσουμε το χρόνο που απαιτεί η εκτεταμένη τοπική βελτιστοποίηση εισαγάγαμε έναν έξυπνο τρόπο επιλογής των προς ανταλλαγή διαδρομών. Επίσης, προσθέσαμε την εκτεταμένη τοπική αναζήτηση για ένα ποσοστό 2% των λύσεων. Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα:

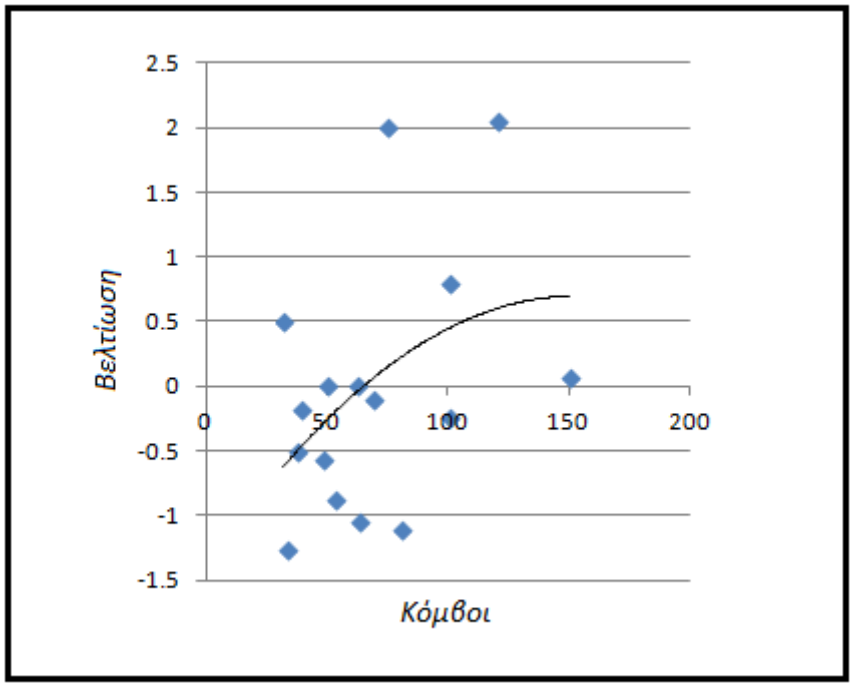
| | **ASR+alpha+extLS** | | **ASR+alpha+clustLS** | | **BKS** | ASR extLS-ASR clustLS |
|---|---|---|---|---|---|---|
| | *Best* | *RPD* | *Best* | *RPD* | | RPD |
| **par1** | 2230.35 | 0 | 2230.35 | 0 | **2230.35** | 0 |
| **par2** | 2441 | 2.064283 | 2393.03 | 0.058537 | **2391.63** | 2.00457161 |
| **par3** | 4165.1 | 2.958407 | 4175.07 | 3.204859 | **4045.42** | -0.238798391 |
| **par4** | 5222.99 | 4.721184 | 5219.34 | 4.648001 | **4987.52** | 0.069932214 |
| **par11** | 7815.05 | 6.823249 | 7657.54 | 4.670258 | **7315.87** | 2.056926898 |
| **par12** | 3694.32 | 3.804525 | 3665.23 | 2.987142 | **3558.92** | 0.793674613 |
| **n32k5** | 2225.59 | 1.532391 | 2214.38 | 1.020985 | **2192** | 0.506236509 |
| **n33k5** | 1717.71 | -3.01834 | 1739.81 | -1.77058 | 1771.17 | -1.270253648 |
| **n69k9** | 3636.66 | -3.83353 | 3640.4 | -3.73463 | 3781.63 | -0.102735963 |

Η τροποποίηση αυτή της τοπικής αναζήτησης έκανε τον αλγόριθμο πολύ πιο σύντομο ωστόσο τα αποτελέσματα ήταν μικτά. Στα προβλήματα που παρουσιάζονται στον πίνακα η μέση βελτίωση ήταν 0,0425% ωστόσο υπάρχουν προβλήματα που η εκτεταμένη αναζήτηση παραμένει καλύτερη. Παρατηρώντας ποια προβλήματα είναι αυτά συμπεραίνεται ότι αποτελούν κατά βάση τα ευκολότερα προβλήματα, ανεξαρτήτως του αριθμού των κόμβων που τα αποτελούν. Με τη λέξη ευκολότερα εννοούμε την ευκολία εύρεσης εφικτών λύσεων όπως αναφέρθηκε και προηγουμένως.

Με αφορμή αυτή την παρατήρηση, ότι το μέγεθος του προβλήματος δεν αποτελεί κριτήριο της βελτίωσης δοκιμάσαμε τους δύο αυτούς αλγορίθμους σε περισσότερα προβλήματα της κατηγορίας An#k#, που αποτελούν ευκολότερα προβλήματα, προκειμένου να εξετάσουμε πως συμπεριφέρονται οι 2 παραλλαγές του αλγορίθμου σε αυτά. Όπως διαπιστώνεται, για αυτά τα προβλήματα ο αλγόριθμος με την εκτεταμένη τοπική αναζήτηση είναι πιο αποδοτικό. Επίσης λόγο του μικρότερου μεγέθους που έχουν πολλά από αυτά τα προβλήματα, ο χρόνος εκτέλεσης του είναι ικανοποιητικός. Παρακάτω παρουσιάζονται τα αποτελέσματα των δυο αλγορίθμων και γίνεται η σύγκριση μεταξύ τους. .

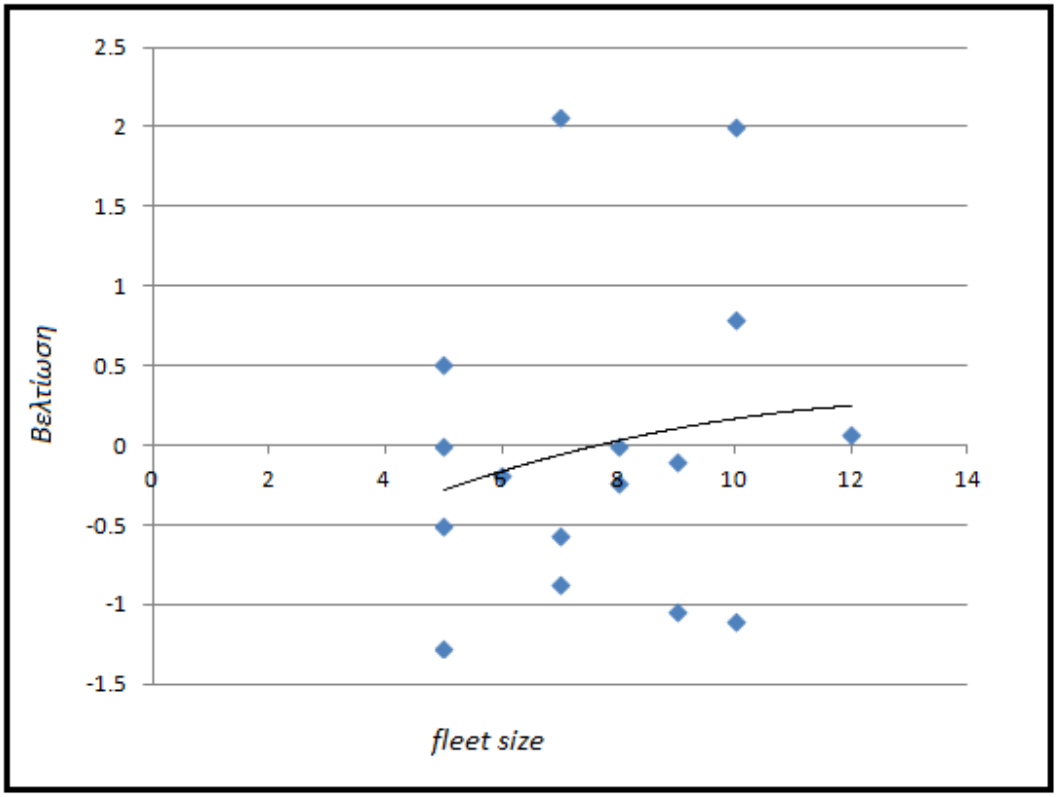
| | **ASR+alpha+extLS** | | **ASR+alpha+clustLS** | | **BKS** | ASR extLS-ASR clustLS |
|---|---|---|---|---|---|---|
| | *Best* | *RPD* | *Best* | *RPD* | | RPD |
| **n37k5** | 2034.04 | -2.14375 | 2044.34 | -1.64822 | **2078.6** | -0.503830087 |
| **n39k6** | 2281.52 | -1.39127 | 2285.71 | -1.21018 | **2313.71** | -0.183312844 |
| **n48k7** | 3109.85 | -4.01549 | 3127.73 | -3.46363 | **3239.95** | -0.571660597 |
| **n53k7** | 3143.62 | -9.18409 | 3171.42 | -8.38098 | **3461.53** | -0.876578946 |
| **n62k8** | 4015.98 | -3.52417 | 4015.98 | -3.52417 | **4162.68** | 0 |
| **n63k9** | 4665.77 | -3.88833 | 4714.83 | -2.87772 | **4854.53** | -1.040546531 |
| **n80k10** | 6119.52 | -4.99572 | 6188.15 | -3.93026 | **6441.31** | -1.10905521 |

Για όλα τα παραπάνω προβλήματα της κατηγορίας An#k# που έγινε η σύγκριση ο αλγόριθμος με την εκτεταμένη τοπική αναζήτηση υπερτερεί σε επιδόσεις της τοπικής αναζήτησης με βάση τις ομάδες. Το παρακάτω διάγραμμα παρουσιάζει το συσχετισμό . παρόλο που δείχνει μια τάση ανόδου της αποτελεσματικότητας όταν αυξάνονται οι κόμβοι, δεν λέει απαραίτητα την αλήθεια. Ο λόγος είναι ότι προβλήματα με παρόμοιο αριθμό κόμβων στις δυο ομάδες προβλημάτων έχουν πολύ μεγάλη διαφορά στη δυσκολία που τα χαρακτηρίζει. Για παράδειγμα το πρόβλημα par2 έχει RPD 2% ενώ το n80k10 -1,20%, αν και διαφέρουν μόνο κατά 5 κόμβους. Όπως είδαμε και στην παραπάνω ανάλυση της σφικτότητας των προβλημάτων, το par2 έχει 97,4% ενώ το n80k10 περίπου 3% λιγότερο και ίσο με 94,2. Επομένως, είναι άλλη μια ένδειξη ότι ο συσχετισμός που παρουσιάζεται στο γράφημα δεν είναι απαραίτητα αληθής.




Πέραν την μελέτης του συσχετισμού της βελτίωσης με τον αριθμό των κόμβων και έχοντας στη διάθεση μας δεδομένα από περισσότερα προβλήματα κρίθηκε σκόπιμο να μελετηθεί ο συσχετισμός της βελτίωσης με τον αριθμό των δρομολογίων που απαιτεί σε κάθε λύση το κάθε πρόβλημα.

| Fleet | RPD |
|---|---|
| 5 | 0 |
| 10 | 2.004572 |
| 8 | -0.2388 |
| 12 | 0.069932 |
| 7 | 2.056927 |
| 10 | 0.793675 |
| 5 | 0.506237 |
| 5 | -1.27025 |
| 9 | -0.10274 |
| 5 | -0.50383 |
| 6 | -0.18331 |
| 7 | -0.57166 |
| 7 | -0.87658 |
| 8 | 0 |
| 9 | -1.04055 |
| 10 | -1.10906 |




Από το παραπάνω διάγραμμα τάσης δεν φαίνεται κάποιος άμεσος συσχετισμός ανάμεσα στον μέγεθος του στόλου που απαιτείται και την βελτίωση. Ωστόσο μπορούμε να εστιάσουμε σε προβλήματα που έχουν ακριβώς τον ίδιο αριθμό οχημάτων και να παρατηρήσουμε ότι για παράδειγμα τα προβλήματα με αριθμό στόλου οχημάτων 5 εμφανίζουν καλύτερα αποτελέσματα με την εκτεταμένη αναζήτηση, σε αντίθεση με τα προβλήματα με αριθμό στόλου 10 που η ομαδοποίηση λειτουργεί καλύτερα. Όμως για μια ακόμα φορά το διάγραμμα είναι παραπλανητικό διότι τα 3 προβλήματα 5 οχημάτων στα οποία η εκτεταμένη αναζήτηση υπερτερεί είναι προβλήματα 32 και 33 κόμβων μικρής δυσκολίας.

Επίσης, στα προβλήματα 10 οχημάτων που υπερτερεί η μέθοδος με τις ομάδες, τα προβλήματα αυτά είναι τα προβλήματα par2(75) και par11(100) ενώ αυτό που υπερτερεί η εκτεταμένη αναζήτηση είναι το An80k10. Παρατηρείται λοιπόν, ένα εκ διαμέτρου αντίθετο αποτέλεσμα στη βελτίωση για δυο προβλήματα φαινομενικά παρόμοια. Επομένως, το συμπέρασμα είναι ότι η τοπολογία του προβλήματος και η δυσκολία του περιορισμού χωρητικότητας που παρουσιάζεται σε καθένα από αυτά παίζει κυρίαρχο ρόλο στον τρόπο που επηρεάζουν οι δυο αυτές προσεγγίσεις τοπικής αναζήτησης τα αποτελέσματα.

Συνοπτικά αποτελέσματα

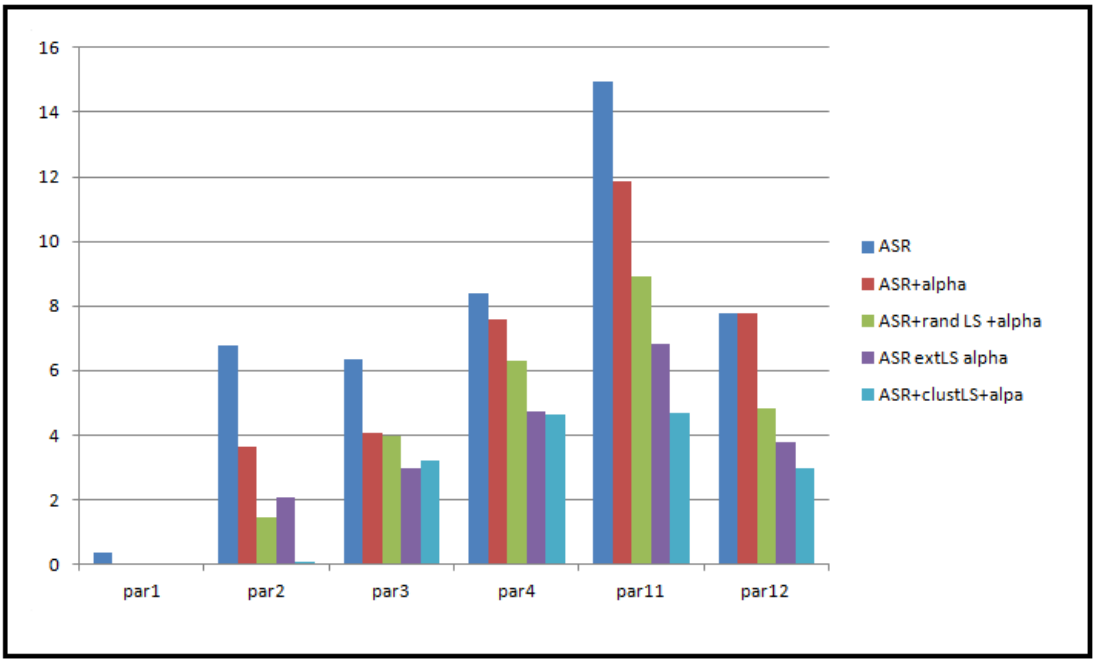
Για την ευκολότερη αντίληψη των αποτελεσμάτων των παραλλαγών του αλγορίθμου AS-Ranked που δημιουργήθηκαν παρουσιάζεται ο πίνακας αποτελεσμάτων. Τα αποτελέσματα είναι τα βέλτιστα που βρέθηκαν μετά από 5 εκτελέσεις σε κάθε πρόβλημα. Οι παράμετροι που χρησιμοποιήθηκαν είναι ίδιοι για όλες τις εκδοχές και ειδικότερα α=1, β=4, ants=Number of nodes, w=6.

| | ASR | | ASR+alpha | | ASR+LS+alpha | | ASR+extLS+alpha | | ASR+LSclusts+alpha | | optimum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **par1** | 2238.02 | 0.343892 | 2230.35 | 0 | 2230.35 | 0 | 2230.35 | 0 | 2230.35 | 0 | **2230.35** |
| **par2** | 2553.8 | 6.780731 | 2478.8 | 3.644794554 | 2426.02 | 1.437931 | 2441 | 2.064283 | 2393.03 | 0.058537 | **2391.63** |
| **par3** | 4301.67 | 6.334324 | 4210.88 | 4.090057398 | 4206.62 | 3.984753 | 4165.1 | 2.958407 | 4175.07 | 3.204859 | **4045.42** |
| **par4** | 5406.7 | 8.404578 | 5364.6 | 7.560470935 | 5302.77 | 6.320777 | 5222.99 | 4.721184 | 5219.34 | 4.648001 | **4987.52** |
| **par11** | 8408.11 | 14.92973 | 8181.72 | 11.83522944 | 7966.35 | 8.891355 | 7815.05 | 6.823249 | 7657.54 | 4.670258 | **7315.87** |
| **par12** | 3835.7 | 7.777078 | 3835.7 | 7.77707844 | 3730.13 | 4.810729 | 3694.32 | 3.804525 | 3665.23 | 2.987142 | **3558.92** |
| **n32k5** | 2278.83 | 3.961223 | 2270.05 | 3.560675182 | 2230.6 | 1.760949 | 2225.59 | 1.532391 | 2214.38 | 1.020985 | 2192 |
| **n33k5** | 1744.98 | -1.47868 | 1739.81 | -1.770581028 | 1739.81 | -1.77058 | 1717.71 | -3.01834 | 1739.81 | -1.77058 | 1771.17 |
| **n69k9** | 3722.47 | -1.5644 | 3689.19 | -2.444448558 | 3657.33 | -3.28694 | 3636.66 | -3.83353 | 3640.4 | -3.73463 | 3781.63 |

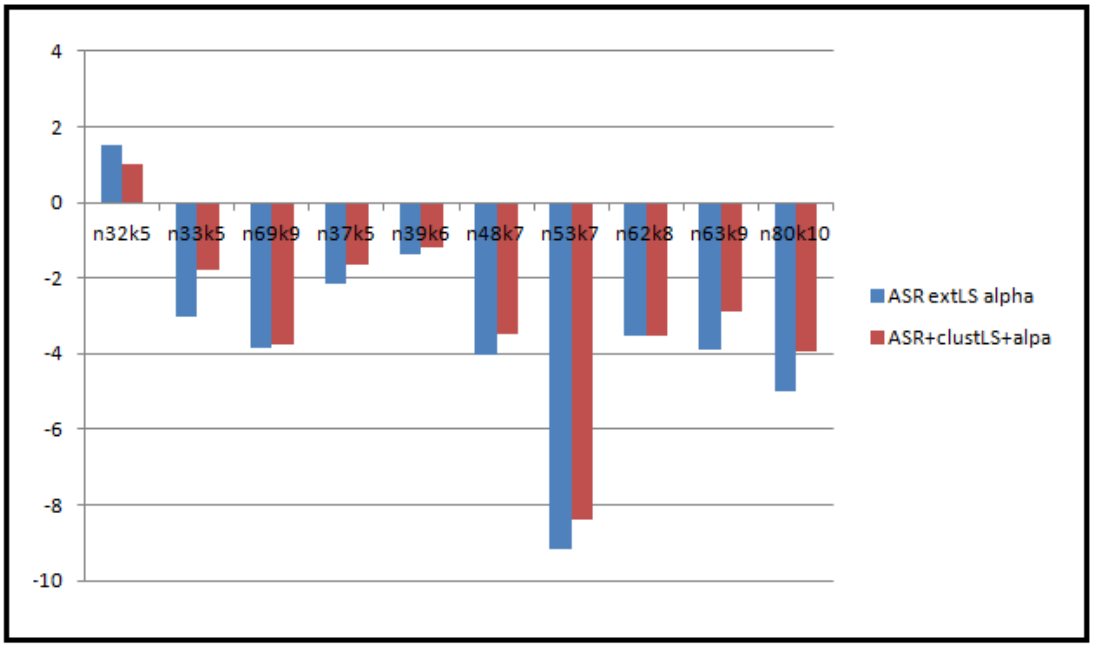
| | ASR+extLS+alpha | | ASR+LSclusts+alpha | | Optimum |
|---|---|---|---|---|---|
| **n37k5** | 2034.04 | -2.14375 | 2044.34 | -1.648224767 | 2078.6 |
| **n39k6** | 2281.52 | -1.39127 | 2285.71 | -1.210177594 | 2313.71 |
| **n48k7** | 3109.85 | -4.01549 | 3127.73 | -3.463633698 | 3239.95 |
| **n53k7** | 3143.62 | -9.18409 | 3171.42 | -8.380976042 | 3461.53 |
| **n62k8** | 4015.98 | -3.52417 | 4015.98 | -3.524171928 | 4162.68 |
| **n63k9** | 4665.77 | -3.88833 | 4714.83 | -2.877724517 | 4854.53 |
| **n80k10** | 6119.52 | -4.99572 | 6188.15 | -3.930256423 | 6441.31 |

Με πράσινο χρώμα υπογραμμίζονται τα βέλτιστα αποτελέσματα που βρέθηκαν για κάθε πρόβλημα. Στην δεξιά στήλη είναι το βέλτιστο αποτέλεσμα της βιβλιογραφίας από τις δημοσιεύσεις [5] για τα προβλήματα Par και [8] για τα προβλήματα An#k#.

Το παρακάτω διάγραμμα δημιουργήθηκε με βάση τον πίνακα αποτελεσμάτων για τα προβλήματα της οικογένειας par και συγκεκριμένα συγκρίνει τις αποκλίσεις RPD από το ολικό βέλτιστο των 5 παραλλαγών του ACO που υλοποιήθηκαν. Η οπτικοποίηση αυτή των διαφορών των αλγορίθμων κάνει ευκολότερη την κατανόηση των διμερών συγκρίσεων που έγιναν προηγουμένως και το μέγεθος της βελτίωσης που υπήρξε μεταξύ της απλής εφαρμογής του αλγορίθμου και των εξελίξεων του. Για μια ακόμη φορά παρατηρούμε ότι προβλήματα που συναντάται η μεγαλύτερη διαφορά μεταξύ απλού AS-Ranked και AS- Ranked με κάποιας μορφής τοπικής αναζήτησης, είναι τα πιο "δύσκολα" προβλήματα και όχι απαραιτήτως αυτά με τον μεγαλύτερο αριθμό κόμβων. Συγκεκριμένα, στο πρόβλημα par2 με σφικτότητα 97,428% το ποσοστό σφάλματος μειώνεται συντριπτικά με την χρήση τοπικής αναζήτησης βάσει ομάδος και στο πρόβλημα par11 με σφικτότητα 98,214% το σφάλμα μειώνεται στο 1/3 σχεδόν του αρχικού.




Επίσης, δημιουργήθηκε διάγραμμα για τα υπόλοιπα προβλήματα που επιλύθηκαν, αυτή τη φόρα όμως συγκρίνοντας τις δύο τελευταίες υλοποιήσεις μεταξύ τους. Ο λόγος που οι τιμές είναι αρνητικές στα περισσότερα από δεδομένα είναι διότι ο αλγόριθμος ξεπερνάει τα βέλτιστα της δημοσίευσης [8] με την οποία έγινε η σύγκριση.



## 4.4 Σύνοψη και προτάσεις

Στην αρχή της μελέτης αυτής θελήσαμε να δούμε κατά πόσο η προσαρμογή των παραμέτρων όπως το ,α, β , w βελτιώνουν το αποτέλεσμα του αλγορίθμου. Στις αρχικές δοκιμές που έγιναν παρατηρήθηκε ότι η αλλαγή του β από 2 έως 5 έχει μικρή σημασία και δεν επηρεάζει ουσιαστικά την επίλυση του προβλήματος. Επίσης η αλλαγή του αριθμού των μυρμηγκιών που εναποθέτουν φερομόνη w επηρεάζει αρνητικά την αποτελεσματικότητα του αλγορίθμου. Για παράδειγμα, δοκιμάσαμε την τιμή w=8 από w=6 και παρόλο που η μέση τιμή των λύσεων βελτιώθηκε, ο αλγόριθμος με w=6 έδινε καλύτερο βέλτιστο αποτέλεσμα. Αυτή η παρατήρηση έρχεται σε συμφωνία με μελέτες όπως η [10] που συμπεραίνει ότι η αλλαγή των παραμέτρων δεν έχουν θετική επίδραση γενικά διότι επηρεάζουν ισορροπία μεταξύ exploration και exploitation.

Αναζητώντας μια αποτελεσματική προσαρμογή που μπορεί να γίνει στις παραμέτρους, η προσοχή εστιάστηκε στο ρόλο που παίζει η παράμετρο alpha για την επιλογή του επόμενου κόμβου. Έτσι, μετά από παρατηρήσεις στο τρόπο με τον οποίο η αλλαγή της τιμής της παραμέτρου αυτής επηρεάζει την αναζήτηση των λύσεων κάνοντας την πιο πλατιά, έγινε προσπάθεια η αλλαγή αυτή να γίνεται κατά τρόπο αποτελεσματικό κατά τη διάρκεια εκτέλεσης του κώδικα. Αποτέλεσμα ήταν η προσαρμογή του alpha με χρήση ασαφούς λογικής, έχοντας ως δεδομένο εισαγωγής τον αριθμό των επαναλήψεων όπου ο αλγόριθμος βρήκε την ίδια βέλτιστη λύση. Τα αποτελέσματα της χρήσης του Fuzzy Alpha, όπως αυτά παρουσιάστηκαν στην προηγούμενη ενότητα, αποδεικνύουν ότι η προσαρμογή της παραμέτρου αυτής βελτιώνει την αποτελεσματικότητα του αλγορίθμου βοηθώντας τα μυρμήγκια να επιλέξουν διαδρομές λιγότερο δημοφιλείς, που χωρίς την αλλαγή του alpha θα παρέμεναν ανεξερεύνητες. Επομένως, η χρήση του Fuzzy Alpha κρίνεται συμφέρουσα λόγο και του αμελητέου υπολογιστικό κόστους που απαιτεί.

Στη συνέχεια προχωρήσαμε στην προσθήκη τοπικής αναζήτησης ώστε να βελτιώσουμε περαιτέρω τα αποτελέσματα. Στην παρουσίαση των αποτελεσμάτων διαπιστώθηκε ότι όλες οι παραλλαγές τοπικής αναζήτησης που χρησιμοποιήθηκαν βελτίωσαν τα αποτελέσματα. Η ευεργετική χρήση αυτών αναλύθηκε προκειμένου να διαπιστωθεί σε ποιές περιπτώσεις υπερτερεί η μία της άλλης εκδοχής. Ως συμπέρασμα εξάγουμε τη ότι την βαρύνουσα σημασία για την αποτελεσματικότητα του 1-1 exchange αλγορίθμου, και κατ' επέκταση της τοπικής βελτιστοποίησης, έχει η επιλογή των διαδρομών που θα ανταλλάξουν τους κόμβους μεταξύ τους.

Στην τοπική αναζήτηση κατά την οποία επιλέγονται τυχαία οι δύο διαδρομές, χαρακτηρίζεται από μεγάλη ταχύτητα, βελτιωμένα αποτελέσματα αλλά όχι εξαιρετικές επιδόσεις. Ο λόγος είναι ότι επαφίεται καθαρά στην τύχη να επιλεγούν 2 διαδρομές που να χρήζουν ανταλλαγής κόμβων. Επομένως, βασιζόμαστε στις πολλές επαναλήψεις και το μικρό χρόνο εκτέλεσης προκειμένου να υπάρξει βελτίωση.

Θέλοντας να αυξήσουμε τις πιθανότητες να επιλεγούν διαδρομές που να αξίζουν να υποστούν 1-1 exchange, προσαρμόσαμε τον αλγόριθμο έτσι ώστε να επιλέγονται με τη σειρά όλοι οι συνδυασμοί μέχρι να μην υπάρχει βελτίωση για έναν αριθμό επιλογών. Ουσιαστικά τυχαία είναι και αυτή η διαδικασία επιλογής, όμως βασιζόμαστε στις πολλαπλές επαναλήψεις προκειμένου να βελτιωθεί το αποτέλεσμα. Πράγμα το οποίο συμβαίνει, κρίνοντας εκ του αποτελέσματος. Ωστόσο αυτή η διαδικασία είναι ιδιαίτερα χρονοβόρα. Ένα επιπλέον αρνητικό αυτού του τρόπου είναι ότι στην περίπτωση που για αρκετούς συνδυασμούς δεν υπάρχει βελτίωση το μόνο που καταφέραμε είναι να σπαταλήσουμε πολύτιμο χρόνο. Συμπερασματικά λοιπόν, είναι μια αποτελεσματική διαδικασία επιλογής αλλά πολύ χρονοβόρα.

Στην επόμενη εξέλιξη της τοπικής αναζήτησης που χρησιμοποιεί ο αλγόριθμος, επιλέξαμε να μη γίνεται τυχαία αλλά να κάνει τις επιλογές των διαδρομών με βάση τις πιο πολλά υποσχόμενες (most promising) ή αλλιώς αυτές που έχουν πιο πολλές πιθανότητες να χρήζουν βελτίωσης, μέσω της ανταλλαγής των κόμβων μεταξύ τους. Η στρατηγική που χρησιμοποιήθηκε είναι ο διαχωρισμός των κόμβων σε ομάδες ανάλογα τη γεωγραφική τους θέση στο νοητό κύκλο με κέντρο την αποθήκη. Η επιλογή του αριθμού των ομάδων δεν βασίζεται σε κάποιο τύπο αλλά θεωρήθηκε ότι πρέπει να είναι ένας αριθμός κοντά στον αριθμό των οχημάτων που είναι απαραίτητα για κάθε πρόβλημα. Πράγματι, αυτή η αλλαγή στη διαδικασία επιλογής βοήθησε τον αλγόριθμο στο να βρει καλύτερα αποτελέσματα για πολλά από τα προβλήματα και κυρίως μείωσε δραματικά το χρόνο που απαιτείται για την εκτέλεση του. Η διαδικασία αυτή παρακάμπτει άσκοπες επιλογές, με ελάχιστες πιθανότητες να είναι καλές, δίνονται προτεραιότητα σε διαδρομές που έχουν απομακρυσμένους μεταξύ τους κόμβους. Ωστόσο, η διαδικασία του most promising βασίζεται στο διαχωρισμό που έχει γίνει στις ομάδες. Επειδή όμως αυτός είναι αυθαίρετος και δίνει μια βασική εικόνα ομαδοποίησης χωρίς να είναι απαραίτητα η καλύτερη, η μέθοδος αυτή είναι πιθανόν να μην διαλέξει τις πραγματικά most promising διαδρομές. Για το λόγο αυτό βλέπουμε και τα μικτά αποτελέσματα στην σύγκριση με την εκτεταμένη τοπική αναζήτηση. Σε μερικά προβλήματα που "βολεύει" η συγκεκριμένη ομαδοποίηση είναι καλύτερη αυτή η μέθοδος ενώ σε άλλα η εκτεταμένη αναζήτηση παρουσιάζει καλύτερα αποτελέσματα. Σε όλες τις περιπτώσεις όμως, ο χρόνος εκτέλεσης του αλγορίθμου με την επιλογή βάση ομάδων είναι συντομότερος.

Άμεση απόρροια των προηγούμενων συμπερασμάτων είναι οι προτάσεις για την περαιτέρω μελέτη τόσο της μεταβολής των παραμέτρων μέσα στον αλγόριθμο όσο και για την διαδικασία τοπικής βελτιστοποίησης που μπορεί να ακολουθηθεί. Στις παραγράφους που ακολουθούν δίνονται κατευθύνσεις για τα μέρη του αλγορίθμου που πιστεύεται ότι αξίζουν περαιτέρω μελέτης και δοκίμων επιπλέον μετατροπών πέραν αυτών που πραγματοποιήθηκαν στην παρούσα εργασία.

Μια διερεύνηση που θα είχε μεγάλο ενδιαφέρον στο κομμάτι της παραμέτρου alpha είναι η εύρεση εναλλακτικών τρόπων προσαρμογής της. Αφού αποδείχθηκε ότι όντως βοηθάει στην αποφυγή της πρόωρης σύγκλισης του αλγορίθμου όταν ο

αλγόριθμος έχει ήδη αρχίσει να συγκλίνει σε ένα ελάχιστο, θα είχε ενδιαφέρον αν αυτή η μεταβολή του μπορεί να λειτουργήσει προληπτικά έχοντας για κριτήριο π.χ. τη διασπορά των λύσεων της εκάστοτε επανάληψης ή τον βαθμό ομοιότητας των λύσεων αυτών. Αυτή η αλλαγή μπορεί να λαμβάνει χώρα σε κάθε επανάληψη όπως περιγράφηκε παραπάνω ή μετά από έναν αριθμό επαναλήψεων λαμβάνοντας υπόψη τη διασπορά όλων των λύσεων που έχουν βρεθεί μέχρι εκείνη τη στιγμή. Σε κάθε περίπτωση, πιστεύεται ότι υπάρχουν περιθώρια βελτίωσης στον τρόπο που προσαρμόζεται η μεταβλητή alpha.

Ο δεύτερος τομέας που χρήζει περαιτέρω μελέτης είναι η τοπική αναζήτηση που λαμβάνει χώρα στον αλγόριθμο. Αυτή μπορεί να εστιαστεί σε 2 μέρη. Πρώτον, στην επιλογή του αλγορίθμου ανταλλαγής. Για παράδειγμα αντί της χρήσης του 1-1 exchange μπορεί να χρησιμοποιηθεί ο 2-2 exchange ή ένας αλγόριθμος relocate όπου δεν ανταλλάσσονται κόμβοι μεταξύ διαδρομών, αλλά μεταφέρονται κόμβοι μόνο από τον ένα στον άλλο. Η δεύτερη προσέγγιση που μπορεί να γίνει είναι στον 1-1 exchange αυτό καθεαυτό. Εκεί, μπορούν να δοκιμαστούν άλλοι τρόποι με του οποίους θα επιλέγονται οι διαδρομές για ανταλλαγή κόμβων. Μια προσαρμογή θα μπορούσε να είναι σε κάθε διαδρομή να υπολογίζονται οι συντεταγμένες του σημείου που αποτελεί το κέντρο βάρος του πολύγονου που σχηματίζει η διαδρομή. Έπειτα να επιλέγονται οι διαδρομές όπου περιέχουν κόμβους που απέχουν μακρύτερα από το κεντρικό σημείο της διαδρομής τους. Μια τέτοια τροποποίηση θα ήταν ανάλογη με την προσέγγιση που έγινε στην ομαδοποίηση της παρούσας εργασίας, ότι αντί για τη διαδρομή με την πιο απομακρυσμένη ομάδα, θα επιλέγαμε τη διαδρομή με τον πιο απομακρυσμένο κόμβο σε σχέση με το κέντρο της.

Μια ακόμα τροποποίηση στην διαδικασία επιλογής των διαδρομών είναι ο πειραματισμός με τις ομάδες αυτές καθαυτές. Δηλαδή, αν μεταβληθεί ο αριθμός τους ή ο τρόπος με τον οποίο χωρίζονται στο γράφημα, αξίζει να μελετηθεί πως μεταβάλλονται τα αποτελέσματα και κατά πόσο μπορεί να βρεθεί ένας βέλτιστος τρόπος ομαδοποίησης ανάλογα με τη τοπολογία του γραφήματος.

## 4.5 Συμπέρασμα

Οι κώδικες που δημιουργήσαμε ικανοποιούν το στόχο της εργασίας, δηλαδή τη εύρεση ενός αποτελεσματικού τρόπου προσαρμογής μιας εκ των παραμέτρων, συγκεκριμένα της παραμέτρου Alpha. Επίσης, μελετήθηκαν 5 παραλλαγές-τροποποιήσεις του αλγορίθμου με στόχο την επίτευξη αποτελεσμάτων που να πλησιάζουν κατά το δυνατό στα ολικά βέλτιστα. Στον πρώτο στόχο, δηλαδή στο κομμάτι της προσαρμογής της παραμέτρου, το αποτέλεσμα ήταν υπεράνω των προσδοκιών καθότι με ελάχιστο υπολογιστικό κόστος καταφέραμε να βελτιώσουμε το αποτέλεσμα σχεδόν στο σύνολο των περιπτώσεων που εξετάστηκαν. Αυτή η μέθοδος προσαρμογής ήταν απλή, υπολογιστικά αποδοτική και πάνω από απ' όλα

αποτελεσματική. Σε μία μόνο περίπτωση το αποτέλεσμα έμεινε το ίδιο, ενώ σε καμία περίπτωση η προσθήκη του μεταβλητού alpha δεν έδωσε χειρότερα αποτελέσματα.

Όσον αφορά το δεύτερο σκέλος, τη βελτίωση των αποτελεσμάτων με στόχο την προσέγγιση των ολικών βέλτιστων, οι υλοποιήσεις είχαν μικτά αποτελέσματα. Στα δύσκολα προβλήματα της οικογένειας par, σε μια περίπτωση ισοφαρίσαμε το βέλτιστο της βιβλιογραφίας. Στις υπόλοιπες περιπτώσεις το σφάλμα κυμάνθηκε από περίπου 0,05% έως περίπου 4,6% για μεγαλύτερα προβλήματα. Στη δεύτερη ομάδα προβλημάτων, τα αποτελέσματα ήταν πέρα για πέρα ενθαρρυντικά. Η σύγκριση των αποτελεσμάτων έγινε με τις βέλτιστες λύσεις της δημοσίευσης [8] που περιλαμβάνει την επίλυση τους με 3 αλγορίθμους, το Γενετικό (GA), τον αλγόριθμο σμήνους σωματιδίων με τελεστές από τον Γενετικό (PSOGA), και τον αλγόριθμο περιορισμένης αναζήτησης (TS). Τα αποτελέσματα του ACO και στις 2 μορφές που τρέξαμε είναι καλύτερα και από τους 3 αυτούς αλγορίθμους και μάλιστα σε ορισμένα προβλήματα η βελτίωση είναι θεαματική.

Από την ανάλυση των αποτελεσμάτων διαπιστώθηκε ότι μεταξύ του αλγορίθμου AS-Ranked εκτεταμένης τοπικής αναζήτησης και αναζήτησης βάσει ομάδος το διαφοροποιητικό στοιχείο των αποτελεσμάτων τους δεν είναι το μέγεθος του προβλήματος, δηλαδή ο αριθμός των κόμβων που περιέχει, αλλά ούτε και ο αριθμός των οχημάτων που απαιτούνται. Η διαφορά στα αποτελέσματα που δίνουν, έγκειται κατά κύριο λόγο στη δυσκολία του προβλήματος το οποίο λύνουν. Με τον όρο δυσκολίας εννοούμε το βαθμό σφικτότητας τους αλλά και τη γενικότερη μορφολογία του γραφήματος που τα περιγράφει. Αυτή η παρατήρηση, αφορά και τα υπόλοιπα ζεύγη συγκρίσεων που πραγματοποιήθηκαν καθώς σε καμία από αυτές δεν παρατηρήθηκε μονότονη συμπεριφορά στη γραμμή τάσης στα αντίστοιχα διαγράμματα.

.

# ΒΙΒΛΙΟΓΡΑΦΙΑ

## Ξένη


*1. Marco Dorigo, Thomas Stützle, "Ant Colony Optimization", The MIT Press,2004, p.71*

*2. Bijaya Keta Panigrahi, Yuhui Shi, Meng-Hiot Lim, "Handbook of swarm Intelligence", Series: Adaptation, Learning, and Optimization, Vol. 8, Springer-Verlag Berlin Heidelberg, 2011*

*3. Amir Salehipour, Kenneth Sörensen, Peter Goos & Olli Bräysy , " An efficient GRASP+VND and GRASP+VNS metaheuristic for the traveling repairman problem", 4OR (June 2011), Volume 9, Issue 2, 189-209*

*4. Glaydston Mattos Ribeiro, Gilbert Laporte, (2012) An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem, Computers & Operations Research 39 728–735*

*5. Ping Chen, Xingye Dong, and Yanchao Niu, (2012) An Iterated Local Search Algorithm for the Cumulative Capacitated Vehicle Routing Problem, H. Tan (Ed.): Technology for Education and Learning, AISC 136, pp. 575–581.*

*6. Sandra Ulrich Ngueveu, Christian Prins, Roberto Wolfler Calvo, "An effective memetic algorithm for the cumulative capacitated vehicle routing problem" Computers & Operations Research 37 (2010) 1877--1885*

*7.James Kennedy , Rusell C. Eberhart "Swarm intelligence", Morgan Kaufmann Publishers,2001*

*8. Ozsoydan Fehmi, Sipahioglu Aydin, "Heuristic Solution Approaches for the Cumulative Capacitated Vehicle Routing Problem", Optimization: A Journal of Mathematical Programming and Operations Research, Volume 62,Issue 10(2013)1321-1340*

*9. Thomas Stützle, Manuel López-Ibáñez, Paola Pellegrini, Michael Maur, Marco Montes de Oca, Mauro Birattari, Marco Dorigo, "Parameter Adaptation in Ant Colony Optimization"Autonomous Search (2012) 191-215*

*10. Paola Pellegrini,Thomas Stützle,Mauro Birattari, "A critical analysis of parameter adaptation in ant colony optimization" Swarm Intelligence, Volume 6, Issue 1, (March 2012) 23-48*


## Ελληνική


*11. Ιωάννης Μαρινάκης , Μυγδάλας Αθανάσιος"Σχεδιασμός και Βελτιστοποίηση της εφοδιαστικής αλυσίδας", Σοφία,2008,σ.471-476*

*12. Μποζάνης Δ. Παναγιώτης,2005, "Αλγόριθμοι", Τζιόλα*

# ΠΑΡΑΡΤΗΜΑ

Αναλυτικά αποτελέσματα των βέλτιστων λύσεων που βρέθηκαν σε κάθε πρόβλημα:

**Par1**
*Route 1:* 0 46 11 38 9 50 16 2 20 35 36 0
*Route 2:* 0 32 1 22 28 31 26 8 48 23 7 43 24 0
*Route 3:* 0 27 6 14 25 13 41 19 42 40 0
*Route 4:* 0 47 18 4 17 37 44 15 45 33 39 0
*Route 5:* 0 12 5 49 10 30 34 21 29 3 0
*Cost: 2230.35*

**Par2**
*Route 1:* 0 75 30 74 28 62 22 21 0
*Route 2:* 0 68 2 73 1 43 41 42 64 0
*Route 3:* 0 4 45 29 48 47 36 69 61 0
*Route 4:* 0 67 46 8 19 54 13 57 15 0
*Route 5:* 0 51 3 44 32 50 18 25 55 0
*Route 6:* 0 17 40 12 72 39 9 31 0
*Route 7:* 0 6 33 63 16 49 24 23 56 0
*Route 8:* 0 7 35 53 14 59 11 0
*Route 9:* 0 26 58 10 38 65 66 0
*Route 10:* 0 34 52 27 5 37 20 70 60 71 0
*Cost: 2393.03*

**Par3**
*Route 1:* 0 53 58 40 21 73 72 74 22 41 75 56 39 25 55 0
*Route 2:* 0 27 69 1 50 76 77 3 79 33 81 78 34 35 71 0
*Route 3:* 0 28 26 12 80 68 29 24 54 4 23 67 0
*Route 4:* 0 52 7 88 62 10 90 32 63 11 19 47 36 49 64 0
*Route 5:* 0 89 18 83 60 5 84 17 45 8 82 48 46 86 0
*Route 6:* 0 6 96 99 59 92 37 98 100 91 85 93 61 16 42 0
*Route 7:* 0 13 94 95 97 87 2 57 15 43 14 44 38 0
*Route 8:* 0 31 70 30 20 51 9 66 65 0
*Cost: 4206.62*

**Par4**
*Route 1:* 0 53 105 26 149 110 4 139 39 56 23 67 0
*Route 2:* 0 112 147 6 94 95 117 97 92 59 61 113 18 0
*Route 3:* 0 146 52 106 7 82 48 124 47 36 143 49 64 46 0
*Route 4:* 0 127 31 88 148 62 10 108 90 126 63 11 107 19 123 0
*Route 5:* 0 138 12 109 80 150 68 121 29 24 134 54 130 55 25 0
*Route 6:* 0 28 50 102 33 81 120 9 135 35 136 71 0
*Route 7:* 0 58 40 21 73 72 74 133 22 75 41 145 115 2 57 15 43 0
*Route 8:* 0 89 60 118 5 84 17 45 125 8 114 83 0
*Route 9:* 0 13 137 87 144 42 142 14 119 44 141 86 0
*Route 10:* 0 27 132 69 1 101 70 30 122 51 103 20 128 131 32 66 0
*Route 11:* 0 96 99 104 93 85 98 37 100 91 16 140 38 0
*Route 12:* 0 111 76 116 77 3 79 129 78 34 65 0
*Cost: 5219.82*

**Par 11**
*Route 1:* 0 18 118 108 109 37 38 39 42 41 44 46 47 49 50 51 48 45 0
*Route 2:* 0 105 106 107 103 67 69 70 71 74 75 72 68 40 43 29 0
*Route 3:* 0 91 90 114 17 16 19 25 24 22 27 30 31 28 32 35 36 34 33 23 26 0
*Route 4:* 0 6 5 3 4 10 9 11 15 14 13 12 8 7 2 1 0
*Route 5:* 0 102 101 99 100 116 98 73 76 77 78 79 80 53 55 58 56 60 63 0
*Route 6:* 0 88 82 111 86 87 92 89 85 112 84 117 113 83 81 119 120 104 21 20 0
*Route 7:* 0 95 96 93 94 97 115 110 52 54 57 59 65 61 62 64 66
*Cost: 7657.54*

**Par 12**
*Route 1:* 0 20 21 22 25 24 27 29 34 36 39 38 0
*Route 2:* 0 23 26 28 30 32 33 31 35 37 0
*Route 3:* 0 67 65 63 62 66 69 64 61 72 74 80 0
*Route 4:* 0 10 13 17 18 19 15 16 14 12 0
*Route 5:* 0 75 1 2 98 96 95 94 97 100 0
*Route 6:* 0 43 47 49 52 50 51 48 45 46 44 42 41 40 68 0
*Route 7:* 0 91 89 88 85 84 83 82 78 76 71 0
*Route 8:* 0 5 3 4 6 9 8 7 11 99 93 92 0
*Route 9:* 0 59 57 55 54 56 58 60 53 0
*Route 10:* 0 90 87 86 81 79 77 73 70 0
*Cost: 3665.23*
---
**n32k5**
*Route 1:* 0 24 14 6 3 2 23 28 0
*Route 2:* 0 30 16 12 1 18 0
*Route 3:* 0 26 7 13 21 31 19 17 0
*Route 4:* 0 27 29 22 9 8 11 4 0
*Route 5:* 0 20 5 25 10 15 0
*Cost: 2214.38*

**n33k5**
*Route 1:* 0 28 18 31 21 14 1 29 0
*Route 2:* 0 4 12 10 30 25 27 0
*Route 3:* 0 2 20 32 13 8 7 26 5 0
*Route 4:* 0 22 15 16 3 9 17 0
*Route 5:* 0 23 11 6 24 19 0
*Cost: 1717.71*

**n37k5**
*Route 1:* 0 15 30 26 18 35 25 8 0
*Route 2:* 0 21 16 22 13 5 33 7 0
*Route 3:* 0 34 36 32 28 31 29 4 12 0
*Route 4:* 0 1 17 14 23 20 19 2 10 6 0
*Route 5:* 0 3 24 9 27 11 0
*Cost: 2034.04*

**n39k6**
*Route 1:* 0 13 30 2 0
*Route 2:* 0 24 3 38 37 31 14 25 0

*Route 3:* 0 26 5 29 28 9 12 35 0
*Route 4:* 0 11 6 1 36 17 23 21 0
*Route 5:* 0 20 32 18 22 34 27 10 16 0
*Route 6:* 0 15 7 8 19 33 4 0
*Cost: 2281.54*

**n48k7**
*Route 1:* 0 14 17 41 2 10 33 0
*Route 2:* 0 18 44 35 32 36 37 3 0
*Route 3:* 0 16 47 28 29 21 30 13 46 0
*Route 4:* 0 40 45 34 9 24 4 42 11 0
*Route 5:* 0 23 43 31 12 5 1 0
*Route 6:* 0 7 27 15 8 39 26 20 0
*Route 7:* 0 6 22 38 19 25 0
*Cost: 3109.85*

**n53k7**
*Route 1:* 0 1 4 28 22 37 2 36 50 43 16 0
*Route 2:* 0 25 39 3 5 14 21 34 0
*Route 3:* 0 31 20 6 33 45 0
*Route 4:* 0 51 8 46 30 44 29 49 10 0
*Route 5:* 0 9 17 41 11 24 52 13 0
*Route 6:* 0 27 35 38 18 40 26 0
*Route 7:* 0 47 7 12 48 42 23 19 15 32 0
*Cost: 3143.62*

**n62k8**
*Route 1:* 0 28 48 56 14 5 60 31 34 25 11 49 35 0
*Route 2:* 0 20 40 6 32 57 50 0
*Route 3:* 0 15 53 12 13 18 30 1 54 9 21 4 0
*Route 4:* 0 52 36 16 37 22 44 2 0
*Route 5:* 0 59 55 26 33 58 17 0
*Route 6:* 0 7 51 10 61 29 43 8 0
*Route 7:* 0 38 19 46 45 47 24 0
*Route 8:* 0 42 3 39 27 23 41 0
*Cost: 4015.98*

**n63k9**
*Route 1:* 0 11 33 53 55 32 58 40 14 0
*Route 2:* 0 15 12 28 50 21 0
*Route 3:* 0 54 29 30 2 5 45 27 0
*Route 4:* 0 25 34 8 59 24 0
*Route 5:* 0 26 19 20 13 60 49 36 48 0
*Route 6:* 0 10 39 23 43 31 38 46 1 0
*Route 7:* 0 44 22 17 35 51 57 0
*Route 8:* 0 56 6 37 52 62 47 4 18 0
*Route 9:* 0 7 16 61 3 41 42 9 0
*Cost: 4665.77*

**n69k9**
*Route 1:* 0 19 58 66 23 13 35 45 0

*Route 2:* 0 31 18 53 9 30 51 3 15 44 6 0
*Route 3:* 0 26 54 5 46 37 50 16 62 0
*Route 4:* 0 28 57 42 14 25 11 63 32 0
*Route 5:* 0 7 67 64 61 21 2 33 49 0
*Route 6:* 0 24 29 40 38 39 56 0
*Route 7:* 0 34 27 65 55 60 4 47 0
*Route 8:* 0 22 12 68 41 20 59 8 0
*Route 9:* 0 43 52 48 1 36 10 17 0
*Cost: 3636.66*

**n80k10**
*Route 1:* 0 73 36 53 64 46 41 15 33 47 0
*Route 2:* 0 7 63 11 34 52 18 0
*Route 3:* 0 51 77 3 74 17 31 27 25 55 0
*Route 4:* 0 40 21 10 71 14 28 79 48 0
*Route 5:* 0 29 59 78 68 16 43 61 57 0
*Route 6:* 0 67 66 70 76 50 45 22 32 4 0
*Route 7:* 0 13 12 44 5 30 6 2 0
*Route 8:* 0 1 62 23 24 37 8 0
*Route 9:* 0 49 38 58 72 54 9 56 69 0
*Route 10:* 0 42 39 60 20 75 19 26 35 65 0
*Cost: 6119.52*