



Πανεπιστήμιο Πειραιώς

Σχολή Βιομηχανικής Διοίκησης και Τεχνολογίας

Μεταπτυχιακό πρόγραμμα: Ειδίκευση Διοίκηση Logistics

Μεταπτυχιακή Διατριβή:

**Επίλυση του προβλήματος δρομολόγησης
ετερογενούς στόλου οχημάτων σταθερού αριθμού με
την χρήση του αλγορίθμου της πυγολαμπίδας**

Ματθόπουλος Παναγιώτης-Πέτρος

Επιβλέπουσα καθηγήτρια: Σοφianoπούλου Στέλλα

Πειραιάς 2015

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή καταπιάνεται με την επίλυση του προβλήματος δρομολόγησης ετερογενούς στόλου οχημάτων σταθερού αριθμού (heterogeneous fixed fleet vehicle routing problem). Για τον σκοπό αυτό αναπτύχθηκε ένας υβριδικός αλγόριθμος πυγολαμπίδας (hybrid firefly algorithm) που ανήκει στους μεθευρετικούς αλγορίθμους νοημοσύνης σμήνους. Ο κλασικός αλγόριθμος της πυγολαμπίδας δημιουργήθηκε για την επίλυση συνεχών προβλημάτων παρ' όλα αυτά με τροποποιήσεις μπορεί να εφαρμοστεί και για την επίλυση διακριτών προβλημάτων όπως είναι το πρόβλημα δρομολόγησης οχημάτων και οι παραλλαγές του.

Το συγκεκριμένο πρόβλημα έχει πρακτικές εφαρμογές καθώς αντιμετωπίζεται καθημερινά από πολλές εταιρίες που ασχολούνται με τον κλάδο των Logistics. Βοηθάει στον τακτικό προγραμματισμό κάθε εταιρίας καθώς καθημερινά οι ανάγκες αποστολής εμπορευμάτων μπορεί να αλλάζουν, τόσο ο αριθμός των πελατών όσο και ο αριθμός των εμπορευμάτων, αλλά ο διατιθέμενος στόλος οχημάτων κάθε εταιρίας παραμένει συνήθως σταθερός. Η συγκεκριμένη παραλλαγή του προβλήματος δρομολόγησης οχημάτων είναι μια από τις παραλλαγές που έχουν μελετηθεί λιγότερο από τους ερευνητές.

Στην διατριβή παρουσιάζεται η μαθηματική μοντελοποίηση του προβλήματος καθώς επίσης και όλα τα σημαντικά βήματα που απαιτούνται για την ανάπτυξη του αλγορίθμου. Για την αξιολόγηση της αποδοτικότητας και αποτελεσματικότητας του αλγορίθμου χρησιμοποιήθηκαν γνωστά προβλήματα, σύνολα δεδομένων, από την βιβλιογραφία τα οποία έλυσε. Επιπλέον ο αλγόριθμος συγκρίνεται με άλλους μεθευρετικούς αλγορίθμους που επιλύουν το ίδιο πρόβλημα.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια Σοφianoπούλου Στέλλα καθώς και τον υποψήφιο διδάκτορα Μητσόπουλο Γιάννη για την πολύτιμη βοήθειά τους και τον χρόνο που αφιέρωσαν για την παρούσα μεταπτυχιακή διατριβή. Επίσης να ευχαριστήσω τους γονείς μου για την συνεχή στήριξη και συμπαράσταση τους καθώς και τον αδερφό μου για την πολύτιμη βοήθεια που μου προσέφερε όποτε την χρειάστηκα.

Περιεχόμενα

Περίληψη	σελ. 2
Ευχαριστίες	σελ. 3
Κεφάλαιο 1: Εισαγωγή	σελ. 6
1.1 Πρόβλημα δρομολόγησης οχημάτων (VRP)	σελ. 6
1.1.1 Πρόβλημα δρομολόγησης οχημάτων με συγκεκριμένο όριο χωρητικότητας (capacitated vehicle routing problem –CVRP)	σελ. 8
1.1.2 Πρόβλημα δρομολόγησης ετερογενών οχημάτων (Heterogeneous fleet vehicle routing problem- HFVRP).....	σελ. 8
1.1.3 Πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (Vehicle routing problem with time windows- VRPTW).....	σελ.8
1.1.4 Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες (Vehicle routing problem with multiple depots-VRPMD).....	σελ.8
1.1.5 Πρόβλημα δρομολόγησης οχημάτων με συλλογές κατά την παράδοση (Vehicle routing problem with pick-up and delivery- VRPPD).....	σελ.8
1.1.6 Πρόβλημα δρομολόγησης οχημάτων με “Backhauls” (Vehicle routing problem with Backhauls- VRPB).....	σελ.9
1.1.7 Στοχαστικό πρόβλημα δρομολόγησης οχημάτων (Stochastic vehicle routing problem- SVRP).....	σελ.9
1.1.8 Πρόβλημα δρομολόγησης οχημάτων με διασπασμένες παραδόσεις (Split delivery vehicle routing problem- SDVRP).....	σελ.9
1.2 Κατηγορίες αλγορίθμων επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης.....	σελ.10
1.2.1 Ακριβείς αλγόριθμοι.....	σελ.10
1.2.2 Ευρετικοί αλγόριθμοι (heuristics).....	σελ.10
1.2.3 Μεθευρετικοί αλγόριθμοι (metaheuristics).....	σελ.10
Κεφάλαιο 2: Βιβλιογραφική ανασκόπηση.....	σελ.11
2.1 Παραλλαγές του προβλήματος δρομολόγησης.....	σελ.11
2.2 Αλγόριθμος της πυγολαμπίδας.....	σελ.14
Κεφάλαιο 3: Αλγόριθμος επίλυσης.....	σελ.16
3.1 Μαθηματική μοντελοποίηση.....	σελ.16
3.2 Αλγόριθμος απληστίας.....	σελ.18
3.3 Αλγόριθμος της πυγολαμπίδας.....	σελ.18
3.3.1 Φωτεινότητα.....	σελ.19
3.3.2 Ελκυστικότητα.....	σελ.19

3.3.3 Απόσταση μεταξύ πυγολαμπίδων.....σελ.20	σελ.20
3.3.4 Μετακίνηση πυγολαμπίδας.....σελ.20	σελ.20
3.3.5 Επιλογή παραμέτρων.....σελ.20	σελ.20
3.4 Αλγόριθμος επίλυσης.....σελ.21	σελ.21
Κεφάλαιο 4: Αποτελέσματα.....σελ.27	σελ.27
Κεφάλαιο 5: Συμπεράσματα.....σελ.38	σελ.38
5.1 Μελλοντικές προεκτάσεις.....σελ.41	σελ.41
Βιβλιογραφία.....σελ.43	σελ.43
Παράρτημα: Απεικόνιση λύσεων των datasetσελ.53	σελ.53

Κεφάλαιο 1: Εισαγωγή

Η διαδικασία διανομής των προϊόντων της εκάστοτε επιχείρησης, δηλαδή η λειτουργία των logistics, αποτελεί μία από τις βασικές δραστηριότητες της εφοδιαστικής αλυσίδας. Είναι μάλιστα η διαδικασία εκείνη η οποία καταλαμβάνει, κατά μέσο όρο, το υψηλότερο ποσοστό στο συνολικό κόστος των δραστηριοτήτων της εφοδιαστικής αλυσίδας (Ballou, 1999). Εφαρμογή του γενικού προβλήματος της μεταφοράς αγαθών και υπηρεσιών από ένα σύνολο οχημάτων για την εξυπηρέτηση πελατών αποτελεί η δρομολόγηση στόλου οχημάτων. Αντικειμενικός στόχος του προβλήματος είναι η ελαχιστοποίηση του κόστους διανομής και εξυπηρέτησης της ζήτησης των πελατών. Για την επίλυση του προβλήματος της διαχείρισης του στόλου οχημάτων έχουν αναπτυχθεί πολλοί αλγόριθμοι και μαθηματικά μοντέλα βελτιστοποίησης που βασίζονται στο μεγαλύτερο ποσοστό τους στο πρόβλημα δρομολόγησης οχημάτων (vehicle routing problem - VRP). Όλοι οι αλγόριθμοι προτείνουν βέλτιστες ή σχεδόν βέλτιστες λύσεις στο πρόβλημα.

1.1 Πρόβλημα δρομολόγησης οχημάτων (VRP)

Το πρόβλημα δρομολόγησης οχημάτων αντιμετωπίζεται καθημερινά από πολλές επιχειρήσεις στην προσπάθειά τους να μειώσουν το κόστος διανομής και συλλογής των προϊόντων τους. Αξίζει να σημειωθεί ότι αποτελεί ένα από τα πιο γνωστά προβλήματα συνδυαστικής βελτιστοποίησης. Το πρώτο άρθρο που δημοσιεύτηκε σχετικά με το πρόβλημα είναι από τους (Dantzig and Ramser, 1959). Γενικεύει το πρόβλημα του πλανόδιου πωλητή (TSP) και για αυτό τον λόγο είναι πρόβλημα πολυπλοκότητας NP-hard (Nondeterministic Polynomial Time Hard) γεγονός που το καθιστά ιδιαίτερα δύσκολο στην επίλυσή του. Μάλιστα όσο αυξάνει το μέγεθος του προβλήματος τόσο πιο δύσκολο είναι να λυθεί σε αποδεκτό υπολογιστικό χρόνο.

Στο κλασικό πρόβλημα του VRP στόχος είναι η ελαχιστοποίηση του κόστους διανομής των προϊόντων στους πελάτες δεδομένου ότι τα οχήματα ξεκινάνε από μία κεντρική αποθήκη και επιστρέφουν σε αυτή έχοντας επισκεφτεί κάθε πελάτη μόνο μία φορά. Η ζήτηση των πελατών, η απόστασή τους από την αποθήκη αλλά και η μεταξύ τους απόσταση, καθώς και η χωρητικότητα των οχημάτων θεωρούνται γνωστά εκ των προτέρων. Αντικειμενικός στόχος της επίλυσης του προβλήματος είναι η εύρεση των καλύτερων δυνατών διαδρομών που ακολουθούν τα φορτηγά για να ελαχιστοποιηθεί το συνολικό κόστος διανομής.

Η μαθηματική μοντελοποίηση του κλασικού προβλήματος του VRP έχει ως εξής:

$$\text{Αντικειμενική συνάρτηση: } \min z = \sum_{i=0}^N \sum_{\substack{j=0 \\ j \neq i}}^N \sum_{k=1}^K c_{ij} x_{ij}^k \quad (1)$$

Υπό τους περιορισμούς:

$$\sum_{i=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall j \in \{1, \dots, N\} \quad (2)$$

$$\sum_{j=0}^N \sum_{k=1}^K x_{ij}^k = 1 \quad \forall i \in \{1, \dots, N\} \quad (3)$$

$$\sum_{j=0}^N x_{ij}^k - \sum_{j=0}^N x_{ji}^k = 0 \quad \forall i \in \{1, \dots, N\}, \quad \forall k \in \{1, \dots, K\} \quad (4)$$

$$\sum_{i=0}^N \sum_{j=0}^N x_{ij}^k d_i \leq Q \quad \forall k \in \{1, \dots, K\} \quad (5)$$

$$\sum_{j=1}^N x_{0j}^k \leq 1 \quad \forall k \in \{1, \dots, K\} \quad (6)$$

$$\sum_{i=1}^N x_{i0}^k \leq 1 \quad \forall k \in \{1, \dots, K\} \quad (7)$$

$$x_{ij}^k \in \{0, 1\} \quad (8)$$

όπου $x_{ij}^k = \begin{cases} 1 & \text{ίσο με 1 αν το όχημα } k \text{ πάει από τον κόμβο } i \text{ στον } j \text{ αλλιώς ίσο με } 0 \\ 0 & \end{cases}$

Η αντικειμενική συνάρτηση αναφέρεται στην ελαχιστοποίηση του συνολικού κόστους μεταφοράς το οποίο εξαρτάται από την συνολική διανυθείσα απόσταση όλων των φορτηγών. Οι περιορισμοί (2) και (3) εκφράζουν ότι κάθε πελάτης επισκέπτεται μόνο μία φορά από το όχημα k . Ο περιορισμός (4) εκφράζει ότι εάν ένα όχημα επισκεφτεί έναν πελάτη πρέπει να φύγει από αυτόν. Ο περιορισμός (5) εκφράζει ότι κανένα φορτηγό δεν μπορεί να εξυπηρετήσει έναν πελάτη του οποίου η ζήτηση υπερβαίνει τη χωρητικότητα του φορτηγού. Οι περιορισμοί (6) και (7) διασφαλίζουν ότι κάθε φορτηγό μπορεί να κάνει μόνο μία διαδρομή. Τέλος, ο περιορισμός (8) δείχνει ότι όλες οι μεταβλητές απόφασης είναι δυαδικές (0 ή 1).

Το κλασικό πρόβλημα του VRP έχει αρκετές υποθέσεις. Βγάζοντας κάποιες από αυτές και βάζοντας περισσότερους περιορισμούς προκύπτουν πολλές παραλλαγές του. Αυτό συμβαίνει επειδή το κλασικό πρόβλημα στερείται πρακτικής εφαρμογής στις επιχειρήσεις. Έτσι ανάλογα με τους περιορισμούς που υπάρχουν κάθε φορά οι βασικότερες παραλλαγές είναι οι εξής:

1.1.1 Πρόβλημα δρομολόγησης οχημάτων με συγκεκριμένο όριο χωρητικότητας (capacitated vehicle routing problem –CVRP)

Στο πρόβλημα αυτό έχουμε έναν δεδομένο αριθμό οχημάτων, που όλα έχουν την ίδια συγκεκριμένη χωρητικότητα, που πρέπει να εξυπηρετήσει τις ήδη γνωστές απαιτήσεις ζήτησης των πελατών από μια κοινή αποθήκη. Το πρόβλημα του CVRP είναι όμοιο με το κλασικό πρόβλημα του VRP με τον πρόσθετο περιορισμό ότι όλα τα οχήματα έχουν ακριβώς την ίδια χωρητικότητα.

1.1.2 Πρόβλημα δρομολόγησης ετερογενών οχημάτων (Heterogeneous fleet vehicle routing problem- HFVRP)

Στο πρόβλημα αυτό έχουμε έναν δεδομένο αριθμό οχημάτων με διαφορετικές χωρητικότητες που πρέπει να εξυπηρετήσουν τις ήδη γνωστές απαιτήσεις ζήτησης των πελατών από μία κοινή αποθήκη.

1.1.3 Πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα (Vehicle routing problem with time windows- VRPTW)

Είναι επέκταση του CVRP καθώς το συγκεκριμένο πρόβλημα περιλαμβάνει ένα χρονικό παράθυρο $[b_i, e_i]$, τόσο για την αποθήκη όσο και για τους πελάτες, κατά την διάρκεια του οποίου ο κάθε πελάτης θα πρέπει να εξυπηρετηθεί.

1.1.4 Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες (Vehicle routing problem with multiple depots-VRPMD)

Το πρόβλημα αυτό γενικεύει την ιδέα της αποθήκης με τέτοιο τρόπο ούτως ώστε να υπάρχουν διαφορετικές αποθήκες από τις οποίες ο κάθε πελάτης μπορεί να εξυπηρετηθεί. Κάθε μία από αυτές διαθέτει τον δικό της στόλο οχημάτων και κάθε όχημα που φεύγει από μια αποθήκη για να εξυπηρετήσει πελάτες πρέπει στο τέλος να επιστρέψει στην ίδια αποθήκη.

1.1.5 Πρόβλημα δρομολόγησης οχημάτων με συλλογές κατά την παράδοση (Vehicle routing problem with pick-up and delivery- VRPPD)

Στο συγκεκριμένο πρόβλημα προβλέπεται η πιθανότητα κάποιοι πελάτες να επιστρέψουν μερικά από τα προϊόντα. Έτσι είναι αναγκαία η πρόβλεψη ύπαρξης απαιτούμενο χώρου στο όχημα για την τοποθέτηση των επιστρεφόμενων προϊόντων.

Αυτό κάνει αρκετά πιο δύσκολη την επίλυση και μπορεί να οδηγήσει στην κακή χρησιμοποίηση της χωρητικότητας των οχημάτων.

1.1.6 Πρόβλημα δρομολόγησης οχημάτων με “Backhauls” (Vehicle routing problem with Backhauls- VRPB)

Στο πρόβλημα αυτό οι πελάτες μπορούν να δεχθούν παραγγελίες ή να επιστρέψουν μέρος των προϊόντων. Η διαφορά με την προηγούμενη περίπτωση είναι ότι τώρα σε κάθε διαδρομή όλες οι παραδόσεις πρέπει να γίνουν πριν από οποιοσδήποτε πιθανές επιστροφές. Οι ποσότητες των προϊόντων που παραδίνονται και επιστρέφονται είναι γνωστές εκ των προτέρων.

1.1.7 Στοχαστικό πρόβλημα δρομολόγησης οχημάτων (Stochastic vehicle routing problem- SVRP)

Στο SVRP ένα ή περισσότερα στοιχεία του προβλήματος, όπως για παράδειγμα ο αριθμός των πελατών ή η ζήτησή τους, δεν θεωρούνται σταθερά, αλλά δυναμικά, δηλαδή δεν είναι γνωστά εκ των προτέρων αλλά μπορούν να αλλάξουν ανά πάσα στιγμή.

1.1.8 Πρόβλημα δρομολόγησης οχημάτων με διασπασμένες παραδόσεις (Split delivery vehicle routing problem- SDVRP)

Στο πρόβλημα αυτό κάθε πελάτης έχει την δυνατότητα να εξυπηρετηθεί από περισσότερα του ενός οχήματα στην περίπτωση που αυτός ο τρόπος εξυπηρέτησης μειώνει το συνολικό κόστος. Τυπική εφαρμογή του είναι όταν ο όγκος παραγγελιών του πελάτη αγγίζει το σύνολο της χωρητικότητας ενός οχήματος. Το πρόβλημα αυτό είναι μια πιο απλή εκδοχή του κλασικού προβλήματος VRP παρόλο που η βέλτιστη λύση του είναι πιο δύσκολο να επιτευχθεί.

Εκτός των ανωτέρω βασικών παραλλαγών του κλασικού προβλήματος δρομολόγησης οχημάτων μπορούν να δημιουργηθούν και άλλες παραλλαγές προσθέτοντας περισσότερους περιορισμούς σε κάθε παραλλαγή. Λεπτομερής περιγραφή του κλασικού προβλήματος δρομολόγησης οχημάτων και των διαφόρων παραλλαγών του προσφέρουν οι (Crainic and Laporte, 1998; Toth and Vigo, 2002).

1.2 Κατηγορίες αλγορίθμων επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης

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

1.2.1 Ακριβείς αλγόριθμοι

Οι ακριβείς αλγόριθμοι εύρεσης της βέλτιστης λύσης ελέγχουν συστηματικά όλες τις πιθανές διαδρομές και επιλέγουν την καλύτερη. Παρότι είναι οι μόνοι αλγόριθμοι που βρίσκουν την βέλτιστη λύση όσο μεγαλώνει το μέγεθος του προβλήματος τόσο αυξάνει και ο χρόνος επίλυσής του φτάνοντας έτσι σε μεγάλα προβλήματα μη αποδεκτό χρόνο επίλυσης του προβλήματος. Μέχρι στιγμής οι καλύτεροι ακριβείς αλγόριθμοι περιορίζονται στο να λύνουν προβλήματα μέχρι 100 πελάτες (Fukasawa et al., 2006). Σε αυτή την κατηγορία αλγορίθμων ανήκουν οι branch and bound και branch and cut αλγόριθμοι.

1.2.2 Ευρετικοί αλγόριθμοι (heuristics)

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

1.2.3 Μεθευρετικοί αλγόριθμοι (metaheuristics)

Οι μεθευρετικοί αλγόριθμοι όπως και οι ευρετικοί δίνουν λύσεις που είναι κοντά στην βέλτιστη λύση. Οι μεθευρετικοί όμως είναι αλγόριθμοι υψηλού επιπέδου, αποδοτικότεροι από τους ευρετικούς (Yang, 2010c) και μπορούν να χρησιμοποιηθούν για την επίλυση σχεδόν όλων των προβλημάτων βελτιστοποίησης (Talbi, 2009). Τα βασικά χαρακτηριστικά τους είναι η εκμετάλλευση (exploitation) και η εξερεύνηση (exploration). Η εκμετάλλευση αφορά την αναζήτηση σε μία μικρή περιοχή στην οποία έχει ήδη εντοπιστεί μία καλή λύση. Η εξερεύνηση σχετίζεται με την αναζήτηση

σε όλο τον χώρο λύσεων με την βοήθεια της τυχαιότητας η οποία δεν επιτρέπει στον αλγόριθμο να εγκλωβίζεται σε τοπικά βέλτιστα. Οι μεθευρετικοί αλγόριθμοι μοντελοποιούν κάποιο φαινόμενο που υπάρχει στην φύση και είναι προσαρμοστικοί. Τέτοιου είδους αλγόριθμοι είναι η προσομοιωμένη απόπτηση - simulated annealing (Osman, 1993), η περιορισμένη αναζήτηση - tabu search (Brandao, 2011; Gendreau et al., 1999), οι γενετικοί και εξελικτικοί αλγόριθμοι - genetic and evolutionary algorithms (Lima et al., 2004), αλγόριθμοι βελτιστοποίησης αποικίας μερμηγκιών - ant colony optimization (Rochat and Taillard, 1995; Li et al., 2009), ο αλγόριθμος της πυγολαμπίδας -firefly algorithm και άλλοι.

Κεφάλαιο 2: Βιβλιογραφική ανασκόπηση

Από την πρώτη φορά που εμφανίστηκε το πρόβλημα δρομολόγησης οχημάτων στην βιβλιογραφία (Dantzig and Ramser, 1959) λόγω της μεγάλης του χρήσης έχει εξελιχτεί και έχουν δημιουργηθεί αρκετές παραλλαγές όπως αναφέρθηκε και στην εισαγωγή. Εξαιτίας αυτού του γεγονότος η βιβλιογραφία για τις διάφορες παραλλαγές του VRP είναι πάρα πολύ μεγάλη. Στο παρών κεφάλαιο θα γίνει μία πολύ σύντομη αναφορά στην βιβλιογραφία που υπάρχει για κάθε παραλλαγή του κλασικού VRP και εκτενέστερη αναφορά στο HFVRP με το οποίο ασχολείται η παρούσα εργασία. Επίσης θα δοθεί έμφαση στην βιβλιογραφία που σχετίζεται με τον αλγόριθμο της πυγολαμπίδας.

2.1 Παραλλαγές του προβλήματος δρομολόγησης

Με το CVRP, που έχει ως μοναδικό περιορισμό την ίδια χωρητικότητα για όλα τα οχήματα, και είναι η πιο κλασική παραλλαγή του προβλήματος ασχολήθηκαν συγγραφείς όπως οι (Rochat and Taillard, 1995; Ralphs et al., 2001; Toth and Vigo, 2002; Ralphs et al., 2003; Fukasawa et al., 2006).

Έμφαση στο VRPTW που απαιτεί η εξυπηρέτηση κάθε πελάτη να ξεκινάει μέσα σε συγκεκριμένο χρονικό παράθυρο έδωσαν συγγραφείς όπως οι (Kolen et al., 1987; Solomon, 1995; Taillard et al., 1997; Cordeau et al., 2001; Cordeau et al., 2002; Lau et al., 2003; Chiang and Russel, 2004).

Με το VRPMD που είναι η παραλλαγή με τις πολλαπλές αποθήκες ασχολήθηκαν οι (Ribeiro and Soumis, 1994; Renaud et al., 1996; Crevier et al., 2007; Yu et al., 2011; Zhang et al., 2011).

Η παραλλαγή του VRPPD έχει μελετηθεί από διάφορους συγγραφείς. Οι (Savelsbergh and Sol, 1995; Xu et al., 2003; Dell'Amico et al., 2007) ασχολήθηκαν με το απλό πρόβλημα του VRPPD ενώ άλλοι συγγραφείς όπως οι (Dumas et al., 1991; Lu and Dessouky, 2006; Ropke and Pisinger, 2006; Ropke et al., 2007; Ropke and Cordeau, 2009; Zachariadis et al., 2009) ασχολήθηκαν με το πρόβλημα προσθέτοντας και χρονικά παράθυρα κάνοντάς το πιο ρεαλιστικό. Για

ολοκληρωμένες ανασκοπήσεις σχετικά με την παραλλαγή του VRPPD μπορούμε να ανατρέξουμε στους (Savelsbergh and Sol, 1998; Cordeau et al., 2008; Parragh et al., 2008)

Με την στοχαστική παραλλαγή του προβλήματος δηλαδή το SVRP ασχολήθηκαν συγγραφείς όπως οι (Gendreau et al., 1996; Laporte and Louveaux, 1998; Yaohuang et al., 2002).

Το VRPPB είναι άλλη μία παραλλαγή του κλασικού VRP στην οποία χρησιμοποιούνται και backhauls. Στην συγκεκριμένη παραλλαγή έδωσαν βάση οι (Goetschalckx and Jacobs-Blecha, 1988; Goetschalckx and Jacobs-Blecha, 1989; Jacobs-Blecha and Goetschalckx, 1992; Goetschalckx and Jacobs-Blecha, 1993; Anily, 1996; Toth and Vigo, 1999; Osman and Wassan 2002; Brandao, 2006).

Το SDVRP είναι άλλη μία παραλλαγή του προβλήματος που επιτρέπει περισσότερα του ενός οχήματα να εξυπηρετούν έναν πελάτη (Dror et al., 1994; Archetti et al., 2001; Chen et al., 2007).

Δεδομένου ότι πολλές επιχειρήσεις κατέχουν έναν ετερογενή στόλο οχημάτων ή νοικιάζουν διαφορετικών ειδών οχήματα για να εξυπηρετήσουν τους πελάτες τους είναι ιδιαίτερα σημαντική η ενασχόληση με την παραλλαγή του HFVRP (Leung et al., 2013). Με το πρόβλημα αυτό που εξυπηρετεί τους πελάτες με διαφορετικού τύπου οχήματα ασχολήθηκαν συγγραφείς όπως οι (Golden et al., 1984; Gendreau et al., 1999; Lima et al., 2004; Prins, 2009; Brandao, 2011). Η παραλλαγή αυτή στην ουσία επιλύει το κλασικό VRP πρόβλημα με ετερογενή στόλο οχημάτων με διαφορετικές χωρητικότητες, σταθερά και μεταβλητά κόστη (Choi and Tcha, 2007; Imran et al., 2009).

Στην βιβλιογραφία έχουν μελετηθεί τρεις διαφορετικές εκδοχές του HFVRP. Οι (Golden et al., 1984) έλαβαν υπόψη ότι τα μεταβλητά κόστη διανέμονται ομοιόμορφα σε όλους τους τύπους οχημάτων καθώς και ότι πάντα υπάρχει διαθεσιμότητα όλων των τύπων των οχημάτων. Στην δεύτερη εκδοχή οι (Gendreau et al., 1999;) θεώρησαν ότι κάθε διαφορετικός τύπος οχήματος έχει διαφορετικά μεταβλητά κόστη και ότι πάντα υπάρχει διαθεσιμότητα όλων των τύπων των οχημάτων. Στην τρίτη εκδοχή οι (Taillard, 1999; Tarantilis et al., 2003; Tarantilis et al., 2004; Li et al., 2007; Li et al., 2010) θεωρούν ότι ο αριθμός των διαθέσιμων οχημάτων κάθε τύπου είναι περιορισμένος. Στην περίπτωση που ο αριθμός των οχημάτων κάθε τύπου είναι απεριόριστος το πρόβλημα ονομάζεται fleet size and mix vehicle routing problem (FSMVRP), ενώ στην περίπτωση που ο αριθμός των διαφορετικών τύπων οχημάτων είναι προκαθορισμένος το πρόβλημα ονομάζεται heterogeneous fixed fleet vehicle routing problem (HFFVRP). Παρότι οι δύο αυτές παραλλαγές είναι παρόμοιες χρησιμοποιούνται σε διαφορετικού τύπου περιπτώσεις. Το FSMVRP είναι πιο κατάλληλο για στρατηγικές αποφάσεις όταν μία εταιρία θέλει να αγοράσει στόλο οχημάτων και πρέπει να καθορίσει το μέγεθός του και την σύστασή του. Από την άλλη το HFFVRP είναι κατάλληλο για λειτουργικές αποφάσεις όταν πρέπει να

αποφασιστεί ποια από τα διαθέσιμα οχήματα θα χρησιμοποιηθούν για την εξυπηρέτηση των πελατών (Brandao, 2011). Πρόσφατα οι (Baldacci et al., 2008) συνοψίζουν τις δύο αυτές βασικές παραλλαγές.

Το HFFVRP έχει πολλές πρακτικές εφαρμογές και συγκριτικά με άλλες παραλλαγές του κλασικού VRP έχει λάβει πολύ μικρότερη προσοχή από τον επιστημονικό κλάδο. Έχουν χρησιμοποιηθεί διάφορες τεχνικές για την επίλυση του προβλήματος οι οποίες είναι οι ακόλουθες. Ο (Taillard, 1999) πρότεινε μια ευρετική μέθοδο παραγωγής στηλών (column generation heuristic method) για την επίλυση του προβλήματος. Αυτή βασίζεται σε μια προσαρμοστική διαδικασία μνήμης (adaptive memory procedure) που χρησιμοποιεί μια ενσωματωμένη περιορισμένη αναζήτηση (tabu search). Για κάθε τύπο οχήματος η μέθοδος αυτή λύνει διαδοχικά ομοιογενή VRPs που έχει ως αποτέλεσμα να υπάρχει ένας μεγάλος αριθμός από διαδρομές οχημάτων. Στην συνέχεια εξασφαλίζεται μια εφικτή λύση για το HFFVRP λύνοντας βέλτιστα ένα ακέραιο γραμμικό πρόβλημα με τέτοιο τρόπο που κάθε μεταβλητή – στήλη είναι μια διαδρομή που παράχθηκε από την εφαρμογή του αλγορίθμου περιορισμένης αναζήτησης.

Οι (Tarantilis et al., 2003; Tarantilis et al., 2004) ανέπτυξαν δύο μεθόδους που ενσωματώνουν την μεθευρετική μέθοδο του ορίου αποδοχής (threshold accepting metaheuristic) και μια απλή ανταλλαγή (swap) καθώς και κινήσεις εξαγωγής-επανεισαγωγής (extraction-reinsertion moves) για να επιτύχουν λύσεις υψηλής ποιότητας. Στην πρώτη μέθοδο κατά την διάρκεια της αναζήτησης δίνονται κάποια σειρά σταθερών ορίων αποδοχής ενώ στην δεύτερη μέθοδο τα όρια αποδοχής αλλάζουν δυναμικά.

Επίσης οι (Li et al., 2007) προσάρμοσαν τον record-to-record travel algorithm που ανέπτυξε ο (Duesk, 1993) για να επιλύσουν το HFFVRP. Σύμφωνα με αυτόν τον αλγόριθμο ανηφορικές κινήσεις επιτρέπονται ούτως ώστε να ξεφύγει ο αλγόριθμος από τα τοπικά βέλτιστα. Δηλαδή η νέα λύση που προκύπτει κάνοντας μια σειρά από τοπικές αναζητήσεις μπορεί να γίνει αποδεκτή ως τωρινή λύση ούτως ώστε να γίνουν και άλλες τοπικές αναζητήσεις κάθε φορά που η αντίστοιχη αντικειμενική συνάρτηση απέχει ένα συγκεκριμένο ποσοστό από την αντικειμενική συνάρτηση της αρχικής λύσης.

Εκτός αυτών οι (Li et al., 2010) πρότειναν για την επίλυση του προβλήματος έναν αλγόριθμο πολλαπλών εκκινήσεων με προσαρμοστική μνήμη (multi-start adaptive memory programming). Σε αυτόν χρησιμοποιείται μια ευριστική μέθοδος για την δημιουργία διαδρομών. Στην συνέχεια ένας τροποποιημένος αλγόριθμος περιορισμένης αναζήτησης χρησιμοποιείται για να βελτιώσει την ποιότητα των λύσεων. Η διαδικασία αυτή επαναλαμβάνεται για συγκεκριμένο αριθμό επαναλήψεων. Τέλος μία διαδικασία βελτιστοποίησης που εφαρμόζει κινήσεις 2- opt και 3- opt εφαρμόζεται στην καλύτερη ευρεθείσα λύση. Επιπλέον η μεθευρετική μέθοδος της επανασύνδεσης μονοπατιών (path-relinking metaheuristic) που

αναπτύχθηκε από τους (Glover and Laguna, 1997) χρησιμοποιείται για την ενίσχυση της απόδοσης του αλγορίθμου.

Πρόσφατα αναπτύχθηκε μια μεθυστική μέθοδος περιορισμένης αναζήτησης για την επίλυση του προβλήματος από τον (Brandao, 2011). Δηλαδή προτάθηκε ένας αλγόριθμος που βασίζεται σε όλα τα βασικά συστατικά της περιορισμένης αναζήτησης που σε μερικές περιπτώσεις οδηγεί σε καλύτερα αποτελέσματα από τα έως ήδη γνωστά.

Επιπλέον οι (Naji-Azimi and Salari, 2013) πρότειναν μια ευρετική μέθοδο ακέραιου γραμμικού προγραμματισμού για την επίλυση του προβλήματος ως συμπληρωματικό εργαλείο για την βελτίωση της απόδοσης των ήδη υπαρχόντων μεθόδων λύσης του HFFVRP.

Τέλος οι (Matei et al., 2015) πρότειναν έναν αποδοτικό μεμετικό αλγόριθμο για την επίλυση του προβλήματος. Αυτός είναι ένας υβριδικός ευρετικός αλγόριθμος που ενσωματώνει έναν γενετικό αλγόριθμο μετανάστευσης (immigration-based genetic algorithm) καθώς και μία διαδικασία τοπικής αναζήτησης.

Σε μια προσπάθεια για να συγκεντρωθεί όλη η βιβλιογραφία και οι τρόποι λύσης του κάθε είδους του VRP οι (Laporte and Osman, 1995; Laporte, 2007; Ball et al., 1995) δημιούργησαν μία εκτενής βιβλιογραφική ανασκόπηση. Επίσης μια επισκόπηση για τις υπάρχον μεθόδους επίλυσης του προβλήματος μπορεί να βρεθεί στους (Laporte, 1992; Golden et al., 1998; Laporte et al., 2000).

2.2 Αλγόριθμος της πυγολαμπίδας

Ένας από τους καινούριους μεθυστικούς αλγορίθμους που έχουν αναπτυχθεί είναι ο αλγόριθμος της πυγολαμπίδας. Ανήκει στην κατηγορία της νοημοσύνης σμήνους και αναπτύχθηκε από τον (Yang, 2008) για να λύνει συνεχή προβλήματα βελτιστοποίησης και κυρίως συνεχή προβλήματα της κατηγορίας NP-hard. Είναι στοχαστικός αλγόριθμος εμπνευσμένος από την φύση και πιο συγκεκριμένα από την συμπεριφορά των πυγολαμπίδων. Είναι αλγόριθμος που βασίζεται σε έναν πληθυσμό λύσεων (population-based). Τέτοιου είδους αλγόριθμοι έχουν αρκετά πλεονεκτήματα σε σχέση με αυτούς που ξεκινάνε αναζήτηση από ένα μόνο σημείο (Prugel-Bennett, 2010). Ο αλγόριθμος της πυγολαμπίδας έχει τρεις κατηγορίες. Υπάρχει ο κλασικός αλγόριθμος που λύνει κυρίως συνεχή προβλήματα και αναπτύχθηκαν υβριδικοί και τροποποιημένοι αλγόριθμοι της πυγολαμπίδας για να επιλύουν και άλλου είδους προβλήματα.

Ο κλασικός αλγόριθμος μπορεί να χρησιμοποιηθεί και για την επίλυση πολυτροπικών (multimodal optimization) προβλημάτων βελτιστοποίησης (Yang, 2009). Επιπλέον με τον κλασικό αλγόριθμο επιλύθηκε το μη γραμμικό πρόβλημα βελτιστοποίησης του σχεδιασμού δοχείου πίεσης (non linear pressure vessel design optimization problem) που είναι συνεχές πρόβλημα (Yang, 2010a).

Η απόδοση του αλγόριθμου της πυγολαμπίδας εξαρτάται σε μεγάλο βαθμό από τις τιμές των σημαντικών παραμέτρων όπως είναι η φωτεινότητα και η ελκυστικότητα των πυγολαμπίδων. Αλλάζοντας τιμές σε αυτές τις παραμέτρους αλλά και χρησιμοποιώντας άλλους τρόπους για τον υπολογισμό τους προκύπτουν τροποποιημένοι αλγόριθμοι. Για την επίλυση μιας σειράς προβλημάτων πολλοί ερευνητές τροποποίησαν τον κλασικό αλγόριθμο σε δυαδικό (Palit et al., 2011; Falcon et al., 2011; Chandrasekaran and Simon, 2012). Επιπλέον οι (Farahani et al., 2011; Yang, 2011a) χρησιμοποίησαν την Γκαουσιανή κατανομή για να αυξήσουν την ταχύτητα σύγκλισης των πυγολαμπίδων. Εκτός αυτών αναπτύχθηκαν αλγόριθμοι πυγολαμπίδας με την χρήση των πτήσεων Levy (Levy flights) για την μετακίνηση των πυγολαμπίδων (Yang, 2010d; Yang, 2011a; Yang, 2012) έχοντας καλύτερα αποτελέσματα από τους γενετικούς αλγορίθμους και τους αλγορίθμους βελτιστοποίησης σμήνους σωματιδίων (particle swarm optimization).

Χρησιμοποιώντας και άλλες μεθευρετικές μεθόδους αρκετοί ερευνητές δημιούργησαν υβριδικούς αλγορίθμους πυγολαμπίδας για την επίλυση αρκετών προβλημάτων (Farahani et al., 2012; Yang et al., 2010; Luthra et al., 2011; Abdullah et al., 2012; Fister et al., 2012; Hassanzadeh et al., 2012).

Παρότι ο αρχικός αλγόριθμος της πυγολαμπίδας αναπτύχθηκε για να λύνει συνεχή προβλήματα βελτιστοποίησης (Yang, 2010a; Yang, 2010d; Yang, 2011a; Yang, 2011b; Yang, 2012; Gandomi et al., 2011) πλέον με τις δύο παραλλαγές του, τους τροποποιημένους και τους υβριδικούς, επιλύει πολλών ειδών προβλήματα.

Με τον αλγόριθμο λύνονται πλέον και πολλά προβλήματα συνδυαστικής βελτιστοποίησης. Οι (Sayadi et al., 2010) δημιούργησαν έναν διακριτό αλγόριθμο πυγολαμπίδας για την ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης όλων των διεργασιών σε προβλήματα χρονικού προγραμματισμού. Οι (Sayadi et al., 2013) δημιούργησαν έναν διακριτό αλγόριθμο πυγολαμπίδας (discrete firefly algorithm – DFA) για την επίλυση διακριτών προβλημάτων. Χρησιμοποίησαν την σιγμοειδή συνάρτηση για να ορίσουν την θέση κάθε πυγολαμπίδας, δηλαδή την ορίζουν με βάση τι πιθανότητες έχει να είναι στην μία θέση ή στην άλλη για να κάνουν τον αλγόριθμο από συνεχή διακριτό. Οι (Kwicien and Filipowicz, 2012) εφάρμοσαν τον αλγόριθμο της πυγολαμπίδας για την βελτιστοποίηση συστημάτων ουρών αναμονής χρησιμοποιώντας την μαθηματική μοντελοποίηση των συστημάτων αυτών.

Εκτός των ανωτέρω ο αλγόριθμος της πυγολαμπίδας και οι παραλλαγές του έχουν χρησιμοποιηθεί για την επίλυση πολλών άλλων προβλημάτων βελτιστοποίησης με διαφορετικούς περιορισμούς και πολλαπλούς στόχους (Lukasik et al., 2009; Abshouri et al., 2011; Chai-ead et al., 2011; Farahani et al., 2011; Gomes, 2012; Nasiri and Meybodi, 2012; Abedinia et al., 2012; Sulaiman et al., 2012; Yang, 2013).

Παρ' όλα αυτά ελάχιστες προσπάθειες έχουν γίνει για την επίλυση του προβλήματος δρομολόγησης οχημάτων με την χρήση του συγκεκριμένου αλγορίθμου. Οι περισσότερες προσπάθειες αφορούν την επίλυση του προβλήματος του πλανόδιου

πωλητή. Οι (Jati and Suyanto, 2011) ανέπτυξαν έναν διακριτό αλγόριθμο πυγολαμπίδας για την επίλυση του συμμετρικού προβλήματος του πλανόδιου πωλητή. Επιπλέον οι (Jati et al., 2013; Kumbharana and Pandey, 2013) ανέπτυξαν έναν εξελικτικό διακριτό αλγόριθμο πυγολαμπίδας για την επίλυση του προβλήματος του πλανόδιου πωλητή χρησιμοποιώντας για την μέτρηση της απόστασης την μέθοδο του Hamming αντί της ευκλείδειας απόστασης. Τέλος όσον αφορά το πρόβλημα δρομολόγησης οχημάτων οι (Pan et al., 2013) χρησιμοποίησαν τον αλγόριθμο της πυγολαμπίδας για την επίλυση του VRPTW χρησιμοποιώντας την κωδικοποίηση που χρησιμοποιείται για το CVRP καθώς και την κωδικοποίηση που χρησιμοποιείται στην βελτιστοποίηση σμήνους σωματιδίων για την μετακίνηση των πυγολαμπίδων.

Λεπτομερής επισκόπηση του αλγορίθμου της πυγολαμπίδας, των παραλλαγών του, των προβλημάτων που μπορεί να επιλύσει καθώς και των εφαρμογών του μπορεί να βρει κανείς στους (Fister et al., 2013; Ali et al., 2014).

Κεφάλαιο 3: Αλγόριθμος επίλυσης

Όπως αναφέρθηκε στην βιβλιογραφική επισκόπηση η παραλλαγή του heterogeneous fixed fleet vehicle routing problem (HFFVRP) έχει πολλές πρακτικές εφαρμογές στις βιομηχανίες καθώς είναι βασικό εργαλείο για την λήψη τακτικών λειτουργικών αποφάσεων που αφορούν την καλύτερη δυνατή επιλογή στόλου οχημάτων για την διανομή προϊόντων με το μικρότερο δυνατό κόστος. Επιπλέον είναι μία από τις παραλλαγές του κλασικού VRP που έχει μελετηθεί λιγότερο σε σχέση με τις υπόλοιπες. Για τους λόγους αυτούς επιλέχθηκε η ανάπτυξη αλγορίθμου για την επίλυση του συγκεκριμένου προβλήματος.

3.1 Μαθηματική μοντελοποίηση

Για την μαθηματική μοντελοποίηση του προβλήματος οι βασικές παραδοχές που λαμβάνουμε υπόψη είναι οι εξής:

1. Για τον υπολογισμό του συνολικού κόστους λαμβάνονται υπόψη τόσο τα μεταβλητά κόστη που εξαρτώνται από την διανυόμενη απόσταση όσο και τα σταθερά κόστη που είναι διαφορετικά για κάθε είδους όχημα
2. Το πρόβλημα υλοποιείται με μία κεντρική αποθήκη
3. Η ζήτηση των πελατών (κόμβων) είναι σταθερή και γνωστή εκ των προτέρων
4. Ο αριθμός των οχημάτων κάθε τύπου είναι γνωστός εκ των προτέρων
5. Οι αποστάσεις μεταξύ των πελατών ανήκουν στον ευκλείδειο δισδιάστατο χώρο

Το μαθηματικό μοντέλο του προβλήματος είναι το εξής:

Αντικειμενική συνάρτηση:

$$\text{Min} \sum_{k \in K} \sum_{j \in N} F_k X_{0j}^k + \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ij}^k \quad (1)$$

Υπό τους περιορισμούς:

$$\sum_{k \in K} \sum_{i \in N} X_{ij}^k = 1 \quad \forall j \in N \quad (2)$$

$$\sum_{i \in N} X_{ij}^k - \sum_{i \in N} X_{ji}^k = 0 \quad \forall k \in K \quad \forall j \in N \quad (3)$$

$$\sum_{j \in N} X_{0j}^k \leq m_k \quad \forall k \in K \quad (4)$$

$$\sum_{i \in N} y_{ij} - \sum_{i \in N} y_{ji} = q_j \quad \forall j \in N \quad (5)$$

$$q_j X_{ij}^k \leq y_{ij} \leq (Q_k - q_i) X_{ij}^k \quad \forall k \in K, \forall i \in N, \forall j \in N \quad (6)$$

$$y_{ij} \geq 0 \quad \forall i \in N, \forall j \in N \quad (7)$$

$$X_{ij}^k \in \{0,1\} \quad \forall i \in N, \forall j \in N, \forall k \in K \quad (8)$$

Όπου $N=(1,2,3,\dots,n)$ είναι οι κόμβοι – πελάτες, k είναι οι τύποι των οχημάτων και $K=(1,2,\dots,m)$ είναι τα οχήματα. F_k είναι το σταθερό κόστος για το όχημα k και C_{ij}^k είναι το μεταβλητό κόστος για το όχημα k που πηγαίνει από τον κόμβο i στον κόμβο j . Q_k είναι η χωρητικότητα του οχήματος k και q_j η ζήτηση του κόμβου j . Επίσης y_{ij} είναι το φορτίο του οχήματος μεταξύ των κόμβων i και j . Τέλος το $X_{ij}^k = \begin{cases} 1 \\ 0 \end{cases}$ είναι ίσο με 1 αν το όχημα k πάει από τον κόμβο i στον j αλλιώς είναι ίσο με 0.

Η αντικειμενική συνάρτηση (1) αναφέρεται στην ελαχιστοποίηση του συνολικού κόστους μεταφοράς το οποίο εξαρτάται από την συνολική διανυθείσα απόσταση όλων των φορτηγών αλλά και από τα σταθερά κόστη κάθε φορτηγού. Ο περιορισμός (2) αναφέρει ότι κάθε πελάτης επισκέπτεται μόνο μία φορά από το όχημα k ενώ ο περιορισμός (3) αναφέρει ότι εάν ένα όχημα επισκεφτεί έναν πελάτη τότε πρέπει να φύγει από αυτόν αναγκαστικά. Ο περιορισμός (4) αναφέρεται στον μέγιστο αριθμό οχημάτων για κάθε είδος οχήματος. Με τον περιορισμό (5) διασφαλίζουμε ότι η διαφορά της χωρητικότητας του οχήματος πριν πάει σε έναν πελάτη με την χωρητικότητα αφού φύγει από αυτόν είναι ίση με την ζήτηση του πελάτη, δηλαδή ότι το όχημα θα διανείμει σε κάθε πελάτη ποσότητα ίση με αυτήν που ζητάει. Με τον περιορισμό (6) εξασφαλίζουμε ότι η μέγιστη χωρητικότητα του οχήματος δεν ξεπερνιέται ποτέ. Τέλος ο περιορισμός (7) δείχνει ότι η μεταβλητή είναι θετική και ο (8) δείχνει ότι η μεταβλητή είναι δυαδική.

Όπως παρατηρείται από την βιβλιογραφική επισκόπηση έχουν αναπτυχθεί διάφορες μέθοδοι για την επίλυση του προβλήματος δρομολόγησης οχημάτων με ετερογενή στόλο οχημάτων με μεταβλητά και σταθερά κόστη. Η προσέγγιση η οποία ακολουθείται στην εν λόγω εργασία αφορά την χρήση του μεθευρετικού αλγορίθμου της πυγολαμπίδας για την εύρεση της βέλτιστης λύσης σε συνδυασμό με έναν αλγόριθμο απληστίας για την εύρεση αρχικών λύσεων.

3.2 Αλγόριθμος απληστίας

Η γενική αρχή των αλγορίθμων απληστίας είναι η ανάπτυξη μυωπικών κατασκευαστικών αλγορίθμων, δηλαδή αλγορίθμων που κατασκευάζουν μια εφικτή λύση για δεδομένο πρόβλημα σε πολυωνυμικό χρόνο ως προς το μέγεθος των δεδομένων. Στην ουσία ακολουθούν την ιδέα πως ό,τι φαίνεται προς στιγμής καλύτερο ως επιλογή υποθέτουν ότι είναι και συνολικά η καλύτερη επιλογή.

Στην εν λόγω εργασία χρησιμοποιήθηκε ένας αλγόριθμος που επιλέγει τυχαία φορτηγά για την εξυπηρέτηση πελατών με τυχαίο τρόπο μέχρι να μην υπάρχει άλλος πελάτης για να εξυπηρετηθεί. Η διαδικασία αυτή επαναλαμβάνεται είκοσι φορές ούτως ώστε να δημιουργήσουμε κάποιες αρχικές λύσεις που χρησιμοποιούνται ως είσοδος στον αλγόριθμο της πυγολαμπίδας για την δημιουργία του αρχικού πληθυσμού λύσεων.

3.3 Αλγόριθμος της πυγολαμπίδας

Ο αλγόριθμος της πυγολαμπίδας είναι μεθευρετικός και ανήκει στην κατηγορία των αλγορίθμων σμήνους και αναπτύχθηκε από τον (Yang, 2008). Ο αλγόριθμος είναι εμπνευσμένος από την συμπεριφορά των πυγολαμπίδων καθώς εκπέμπουν λάμψεις και έλκονται από αυτές (Yang, 2010a). Ο συγκεκριμένος αλγόριθμος ακολουθεί τρεις βασικούς κανόνες για την εφαρμογή του (Yang, 2010b). Αυτοί είναι οι εξής:

- Οι πυγολαμπίδες είναι αδιάφορου φύλου. Μία πυγολαμπίδα μπορεί να προσελκύσει οποιαδήποτε άλλη χωρίς να εξετάζεται το φύλο της.
- Η ελκυστικότητα των πυγολαμπίδων είναι ανάλογη της φωτεινότητάς τους. Η φωτεινότητα μιας πυγολαμπίδας μειώνεται όσο αυξάνεται η απόστασή της από άλλες πυγολαμπίδες. Η πυγολαμπίδα κινείται προς άλλη φωτεινότερη πυγολαμπίδα. Μια πυγολαμπίδα δεν μπορεί να προσελκύσει μία άλλη που είναι φωτεινότερη αλλά μόνο μπορεί να προσελκυστεί από φωτεινότερη.
- Η φωτεινότητα της πυγολαμπίδας εξαρτάται από την τιμή της αντικειμενικής συνάρτησης. Για προβλήματα μεγιστοποίησης η φωτεινότητα είναι ανάλογη της τιμής της αντικειμενικής συνάρτησης. Αντίθετα για προβλήματα ελαχιστοποίησης είναι αντιστρόφως ανάλογη.

Για να ξεκινήσει να λειτουργεί ο αλγόριθμος απαιτεί την ύπαρξη ενός αρχικού πλήθους πυγολαμπίδων κάθε μια από τις οποίες αναπαριστά σε μορφή διανύσματος

μια λύση του προβλήματος. Για να δημιουργήσουμε τον αρχικό αυτό πληθυσμό των πυγολαμπίδων, δηλαδή για να παράξουμε κάποιες αρχικές λύσεις που θα είναι η εκκίνηση του αλγορίθμου χρησιμοποιήθηκε ένας αλγόριθμος απληστίας.

Οι βασικότεροι παράγοντες του αλγορίθμου της πυγολαμπίδας είναι η φωτεινότητα της πυγολαμπίδας, η ελκυστικότητα, η απόσταση μεταξύ δύο πυγολαμπίδων και ο τρόπος με τον οποίο γίνεται η μετακίνηση της μίας πυγολαμπίδας προς την άλλη.

3.3.1 Φωτεινότητα

Η φωτεινότητα ελαττώνεται όσο απομακρύνεται από την πηγή του φωτός, επομένως πρέπει να επιτρέπεται να ποικίλει ανάλογα με τον βαθμό απορρόφησής της (Yang, 2008). Στην απλούστερή της μορφή η φωτεινότητα ποικίλει σύμφωνα με τον νόμο του αντίστροφου τετραγώνου:

$$I(r) = \frac{I_s}{r^2} \quad (3.3.1)$$

Όπου I_s είναι η φωτεινότητα στην πηγή του φωτός. Η φωτεινότητα I ποικίλει ανάλογα με την απόσταση (r) και ενός σταθερού συντελεστή απορρόφησης του φωτός (γ). Η εξίσωση για την φωτεινότητα γίνεται:

$$I = I_0 \times e^{-\gamma r} \quad (3.3.2)$$

Όπου I_0 είναι η αρχική φωτεινότητα. Με βάση αυτά παρατηρούμε ότι η φωτεινότητα I είναι σχετική καθώς εξαρτάται από την απόσταση ενώ η αρχική φωτεινότητα I_0 είναι απόλυτη. Για να αποτραπεί η μοναδικότητα στο $r=0$ στην μαθηματική έκφραση I_s/r^2 χρησιμοποιείται εν τέλει η Γκαουσιανή μορφή για την έκφραση της φωτεινότητας (Yang, 2010d) :

$$I = I_0 \times e^{-\gamma r^2} \quad (3.3.3)$$

Επιπλέον η αρχική φωτεινότητα (I_0) είναι αντιστρόφως ανάλογη της αντικειμενικής συνάρτησης του προβλήματος. Δηλαδή $I_0 = 1/z$, όπου z είναι η αντικειμενική συνάρτηση.

3.3.2 Ελκυστικότητα

Καθώς η ελκυστικότητα μιας πυγολαμπίδας είναι ανάλογη της φωτεινότητας που διακρίνει μια γειτονική πυγολαμπίδα μπορεί να οριστεί με τον ακόλουθο τρόπο (Yang, 2009):

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \quad (3.3.4)$$

όπου το β_0 είναι η ελκυστικότητα στο σημείο $r=0$. Παρατηρούμε ότι η ελκυστικότητα (β) είναι σχετική, ενώ το β_0 είναι απόλυτο.

3.3.3 Απόσταση μεταξύ πυγολαμπίδων

Στον αλγόριθμο η απόσταση μεταξύ δύο πυγολαμπίδων i και j υπολογίζεται χρησιμοποιώντας τον Καρτεσιανό τρόπο υπολογισμού της απόστασης. Δηλαδή είναι:

$$r_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (3.3.5)$$

Στον δισδιάστατο χώρο η εξίσωση (3.3.5) είναι της μορφής:

$$r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.3.6)$$

3.3.4 Μετακίνηση πυγολαμπίδας

Ο τρόπος με τον οποίο μία πυγολαμπίδα i κινείται προς μία φωτεινότερη j που την ελκύει ακολουθεί την παρακάτω εξίσωση:

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 \times e^{-\gamma r_{ij}^2} \times (\mathbf{x}_i - \mathbf{x}_j) + \alpha \times (\text{rand} - 1/2) \quad (3.3.7)$$

Στην παραπάνω εξίσωση ο δεύτερος όρος οφείλεται στην ελκυστικότητα ενώ ο τρίτος όρος είναι τυχαίος αριθμός με το α να είναι ο παράγοντας της τυχειότητας και το rand μια μηχανή γέννησης τυχαίων αριθμών ομοιόμορφα κατανομημένων στο διάστημα $[0,1]$. Αξίζει να σημειωθεί ότι λόγω του τρίτου όρου η εξίσωση (3.3.7) μπορεί εν μέρει να είναι και τυχαίος περίπατος εφόσον η μετακίνηση δεν γίνεται κατευθυνόμενα προς ένα συγκεκριμένο σημείο, δηλαδή εάν δεν συγκλίνουν όλες οι πυγολαμπίδες προς ένα βέλτιστο σημείο. Στην περίπτωση που το $\beta_0=0$ τότε η εξίσωση (3.3.7) γίνεται ένας απλός τυχαίος περίπατος (Arora and Singh, 2013).

3.3.5 Επιλογή παραμέτρων

Στον αλγόριθμο της πυγολαμπίδας πολύ σπουδαίο ρόλο για την καλή και αποτελεσματική λειτουργία του παίζουν κάποιες από τις παραμέτρους. Αυτές αφορούν τον αριθμό των πυγολαμπίδων (n), τον παράγοντα (γ), τον παράγοντα (β_0) και τον παράγοντα (α).

Ο σημαντικότερος παράγοντας είναι ο (γ) που είναι ο συντελεστής απορρόφησης του φωτός. Η τιμή του καθορίζει σε μεγάλο βαθμό την ταχύτητα σύγκλισης του αλγορίθμου. Για τις περισσότερες εφαρμογές ο παράγοντας παίρνει τιμές μεταξύ $[0.1, 10]$ (Imanirad et al., 2013).

Επιπλέον την ταχύτητα σύγκλισης του αλγορίθμου επηρεάζει και ο αριθμός των πυγολαμπίδων, δηλαδή ο αριθμός των λύσεων που επεξεργάζεται ο αλγόριθμος. Έχει αποδειχτεί ότι ο αλγόριθμος συγκλίνει αρκετά γρήγορα όταν ο αριθμός των πυγολαμπίδων (n) είναι μεταξύ 20 και 50 (Yang, 2009).

Εκτός αυτών ο παράγοντας β_0 για τις περισσότερες περιπτώσεις είναι ίσος με 1 ($\beta_0=1$) και ο παράγοντας $\alpha \in [0,1]$ σύμφωνα με τον (Yang, 2009).

Δύο ασυμπτωτικές περιπτώσεις εμφανίζονται όταν ο παράγοντας $\gamma \rightarrow 0$ και όταν $\gamma \rightarrow \infty$. Όταν το $\gamma \rightarrow 0$ τότε η ελκυστικότητα είναι σταθερή και $\beta=\beta_0$ που ισοδυναμεί με φωτεινότητα που δεν μειώνεται ποτέ. Δηλαδή τότε μία πυγολαμπίδα θα ήταν εμφανής σε όλες τις υπόλοιπες πυγολαμπίδες και άρα ένα ολικό βέλτιστο μπορεί εύκολα να βρεθεί. Αντιθέτως όταν το $\gamma \rightarrow \infty$ τότε η ελκυστικότητα όλων των πυγολαμπίδων είναι μηδενική. Στην περίπτωση αυτή οι πυγολαμπίδες κινούνται τελείως τυχαία και έχουμε έναν τελείως τυχαίο περίπατο.

3.4 Αλγόριθμος επίλυσης

Ο αλγόριθμος που αναπτύχθηκε για την επίλυση του προβλήματος βασίστηκε στον αλγόριθμο της πυγολαμπίδας. Στο παρών υποκεφάλαιο παρουσιάζεται ο τρόπος με τον οποίο αναπτύσσεται ο αλγόριθμος βήμα βήμα.

Βήμα 1:

Στο πρώτο βήμα του αλγορίθμου χρησιμοποιείται ένας αλγόριθμος απληστίας ούτως ώστε να δημιουργήσουμε κάποιες αρχικές λύσεις. Ο λόγος που γίνεται αυτό είναι ότι ο αλγόριθμος της πυγολαμπίδας για να ξεκινήσει πρέπει να έχει έναν αρχικό πληθυσμό πυγολαμπίδων, δηλαδή έναν αρχικό πληθυσμό λύσεων. Κάθε λύση του αλγορίθμου απληστίας είναι και μία πυγολαμπίδα. Ο αλγόριθμος απληστίας επιλέγει τυχαία ένα φορτηγό και πηγαίνει σε κάθε κόμβο τυχαία μέχρι να μην έχει επαρκής χωρητικότητα για να ικανοποιήσει την ζήτηση των κόμβων. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να εξυπηρετηθούν όλοι οι κόμβοι του προβλήματος. Με τον τρόπο αυτό ο αλγόριθμος απληστίας τρέχει μέχρι να δώσει είκοσι διαφορετικές μεταξύ τους λύσεις. Κάθε λύση αναπαρίσταται με ένα διάνυσμα που περιλαμβάνει όλες τις διαδρομές που ακολουθήθηκαν, δηλαδή την σειρά με την οποία επισκεφτήκαν οι κόμβοι και την σειρά με την οποία χρησιμοποιήθηκαν τα φορτηγά. Όλες οι λύσεις υπολογίζουν το κόστος της συνολικής διαδρομής, δηλαδή υπολογίζουν την τιμή της αντικειμενικής συνάρτησης αφού έχουν επισκεφτεί όλοι οι κόμβοι.

Βήμα 2:

Το δεύτερο βήμα του αλγορίθμου αφορά την επιλογή των τιμών των παραμέτρων που χρησιμοποιούνται στον αλγόριθμο της πυγολαμπίδας. Για το συγκεκριμένο πρόβλημα θέτουμε τις εξής τιμές στις παραμέτρους: $\beta_0=1$, $\gamma=1$, $\alpha=0,2$, $n=20$ και $iterations=1000$. Το n είναι ο αριθμός των πυγολαμπίδων και το $iterations$ σημαίνει ότι η διαδικασία που θα αναπτυχθεί στην συνέχεια επαναλαμβάνεται 1000 φορές.

Βήμα 3:

Στο συγκεκριμένο βήμα οι λύσεις που έχουμε πάρει από τον αλγόριθμο απληστίας μετατρέπονται σε πυγολαμπίδες. Η λύση, δηλαδή η πυγολαμπίδα, αναπαριστάται με ένα διάνυσμα μεγέθους $n+k-1$. Ο τρόπος με τον οποίο γίνεται η μετατροπή ακολουθεί την κωδικοποίηση που χρησιμοποιούν οι (Pan et al., 2013) που είναι και το μόνο επιστημονικό άρθρο ως τώρα που λύνει μια παραλλαγή του VRP με την χρήση του αλγορίθμου της πυγολαμπίδας. Η λύση του αλγορίθμου απληστίας είναι μια ακολουθία κόμβων στους οποίους παρεμβάλλεται η αποθήκη όποτε το όχημα πρέπει να γυρίσει πίσω. Με την κωδικοποίηση των (Pan et al., 2013) η αποθήκη αντικαθίσταται με έναν οριοθέτη ο οποίος δηλώνει επίσης ποιο φορτηγό χρησιμοποιήθηκε. Το διάνυσμα είναι μεγέθους $n+k-1$ γιατί στο τέλος του διανύσματος δεν βάζουμε οριοθέτη. Η κάθε λύση έχει την μορφή του σχεδιαγράμματος 3.1. Σε αυτό έχουμε μία πυγολαμπίδα με 8 κόμβους και 3 φορτηγά. Τα φορτηγά αναπαρίστανται με τους έντονους αριθμούς (9 και 10) ενώ το τελευταίο φορτηγό δεν απεικονίζεται στην λύση καθώς στο τέλος του διανύσματος δεν βάζουμε οριοθέτη.

Σχεδιάγραμμα 3.1: Αναπαράσταση πυγολαμπίδας με 8 κόμβους και 3 φορτηγά.

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

Βήμα 4:

Στο βήμα αυτό υπολογίζουμε τις σχετικές φωτεινότητες I και τις σχετικές ελκυστικότητες β για όλες τις πυγολαμπίδες. Η φωτεινότητα της κάθε πυγολαμπίδας υπολογίζεται από την εξίσωση (3.3.3) ενώ η ελκυστικότητα από την εξίσωση (3.3.4).

Βήμα 5:

Στο βήμα αυτό ξεκινάει ο κυρίως αλγόριθμος της πυγολαμπίδας. Αρχικά για κάθε πυγολαμπίδα συγκρίνουμε την φωτεινότητά της σε σχέση με τις φωτεινότητες των υπολοίπων πυγολαμπίδων. Η λιγότερο φωτεινή πυγολαμπίδα ελκύεται από την περισσότερη φωτεινή πυγολαμπίδα και μετακινείται προς αυτήν. Η μετακίνηση αυτή είναι η ανανέωση της θέσης της πυγολαμπίδας (Position update).

Η ανανέωση της θέσης της πυγολαμπίδας γίνεται με βάση την εξίσωση (3.3.7). Από την κωδικοποίηση αυτή προκύπτει η νέα πυγολαμπίδα εκφρασμένη σε πραγματικούς

αριθμούς. Στην συνέχεια αποκωδικοποιούμε την πυγολαμπίδα ούτως ώστε να περιλαμβάνει ακέραιους αριθμούς για να είναι εκφρασμένη όπως είναι και οι αρχικές λύσεις. Η αποκωδικοποίηση αυτή γίνεται ακολουθώντας τον τρόπο που προτείνεται από τους (Pan et al., 2013). Βαθμονομούμε το διάνυσμα που προκύπτει και αντιστοιχίζουμε τις βαθμονομημένες τιμές, δηλαδή τις ακέραιες τιμές, στις πραγματικές. Με τον τρόπο αυτό δημιουργείται η καινούρια πυγολαμπίδα. Η όλη διαδικασία φαίνεται στα σχεδιαγράμματα 3.2 έως 3.5.

Σχεδιάγραμμα 3.2: Κωδικοποίηση της πυγολαμπίδας σε πραγματικούς αριθμούς

0.1	0.4	0.2	0.6	0.9	0.3	0.5	1	0.7	0.8
-----	-----	-----	-----	-----	-----	-----	---	-----	-----

Σχεδιάγραμμα 3.3: Η λύση που προκύπτει μετά τις πράξεις του position update

0.35	0.28	0.1	0.9	0.73	0.86	0.57	0.4	0.6	1
------	------	-----	-----	------	------	------	-----	-----	---

Σχεδιάγραμμα 3.4: Βαθμονόμηση της λύσης μετά το position update

0.1	0.28	0.35	0.4	0.57	0.6	0.73	0.86	0.9	1
-----	------	------	-----	------	-----	------	------	-----	---

Σχεδιάγραμμα 3.5: Αποκωδικοποιημένη λύση σε ακέραιους αριθμούς μετά την αντιστοίχιση των βαθμονομημένων τιμών

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

Η λύση του σχεδιαγράμματος 3.5 είναι η τελική λύση που προκύπτει μετά από την διαδικασία που αναλύθηκε προηγουμένως.

Στην συνέχεια ακολουθούν κάποιιοι έλεγχοι. Ελέγχουμε αν η καινούρια πυγολαμπίδα που δημιουργήθηκε μετά την ανανέωση της θέσης της είναι μοναδική στο σύνολο των λύσεων. Επίσης ελέγχουμε εάν υπάρχει οριοθέτης στην αρχή του διανύσματος. Αν γίνεται κάτι τέτοιο τότε ο οριοθέτης μετακινείται στο τέλος του διανύσματος. Επιπλέον ελέγχουμε εάν υπάρχει δύο οριοθέτες ή περισσότεροι μαζί στην σειρά. Εάν γίνεται κάτι τέτοιο τότε κρατείται ο πρώτος οριοθέτης στην θέση που βρίσκεται και οι υπόλοιποι που ακολουθούν μετακινούνται στο τέλος του διανύσματος. Τέλος χρησιμοποιείται ένα penalty factor για τον έλεγχο της χωρητικότητας. Το penalty factor είναι ένας πολύ μεγάλος αριθμός που προστίθεται στην αντικειμενική συνάρτηση. Με την χρήση του η αντικειμενική συνάρτηση έχει ως εξής:

$$\text{Min} \sum_{k \in K} \sum_{j \in N} F_k X_{0j}^k + \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ij}^k + R * \sum_{k=1}^K \max(\sum_{i=1}^N Q_k y_{ik} - q, 0) \quad (3.4.1)$$

Για να γίνει ο έλεγχος ελέγχουμε εάν η χωρητικότητα του φορτηγού καλύπτει την ζήτηση των κόμβων που εξυπηρετεί.

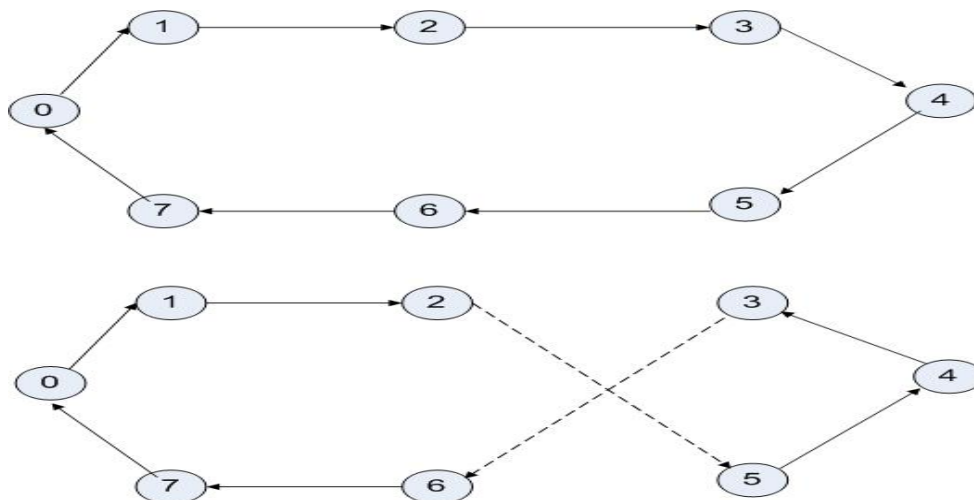
Βήμα 6:

Σε περίπτωση που η ζήτηση καλύπτεται το penalty factor είναι ίσο με μηδέν επομένως δεν επηρεάζεται η αντικειμενική συνάρτηση.

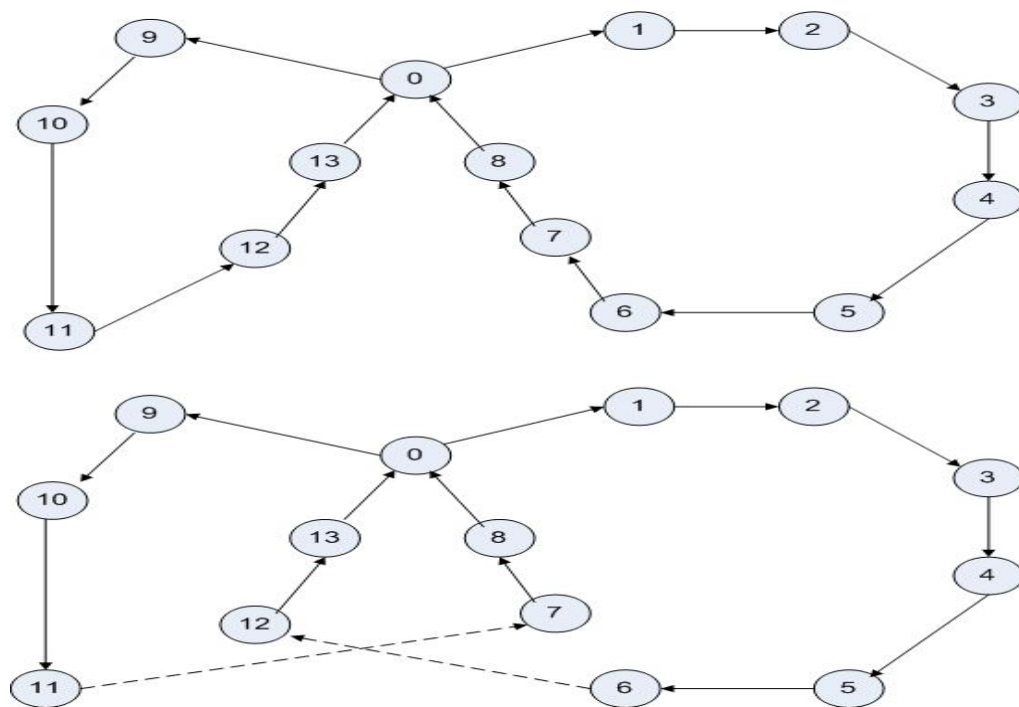
Στην περίπτωση που το penalty factor είναι διάφορο του μηδενός αναδιανέμουμε τους κόμβους στην διαδρομή αφαιρώντας κόμβους που δεν μπορούν λόγω χωρητικότητας να εξυπηρετηθούν από ένα φορτηγό τοποθετώντας τους σε άλλο φορτηγό με επαρκή χωρητικότητα. Κατά την διαδικασία αυτή επιδιώκουμε πάντα την ελάχιστη αύξηση ή μέγιστη μείωση του κόστους της συνολικής διαδρομής.

Αν αυτή η διαδικασία είναι επιτυχής τότε γίνεται μια διαδικασία τοπικής αναζήτησης που ονομάζεται 2-opt με σκοπό να βελτιωθεί η λύση. Η μέθοδος αποτελείται από δύο διαδικασίες. Η πρώτη είναι η intra-route κατά την οποία διαγράφονται δύο ακμές μέσα σε μία υποδιαδρομή της λύσης, δηλαδή σε διαδρομή που κάνει ένα από τα φορτηγά της λύσης, και δημιουργούνται δύο νέες ακμές με στόχο την μείωση του κόστους της υποδιαδρομής, δηλαδή της συνολικά διανυόμενης απόστασης. Στην συγκεκριμένη διαδικασία δεν λαμβάνεται υπόψη το κριτήριο της χωρητικότητας καθώς η χωρητικότητα μέσα στην ίδια διαδρομή δεν μπορεί να αλλάξει. Η δεύτερη διαδικασία είναι η inter-route κατά την οποία διαγράφεται μία ακμή από μία υποδιαδρομή της λύσης και μία ακόμα ακμή από μία άλλη υποδιαδρομή της λύσης. Στην συνέχεια δημιουργούνται δύο νέες ακμές ενώνοντας κόμβους από τις διαφορετικές υποδιαδρομές αλλάζοντας στην την συνολική λύση. Η διαδικασία αυτή όπως και η intra-route γίνεται με στόχο την μείωση του κόστους της συνολικής λύσης. Στην περίπτωση όμως της inter-route γίνεται και έλεγχος της χωρητικότητας της νέας λύσης που προκύπτει ούτως ώστε να είναι έγκυρη και εφικτή η λύση.

Σχεδιάγραμμα 3.6: 2-opt intra route



Σχεδιάγραμμα 3.7: 2-opt inter route



Για τις λύσεις αυτές γίνονται και οι δύο διαδικασίες της μεθόδου 2-opt. Αρχικά γίνεται το intra-route για κάθε ζεύγος ακμών κάθε υποδιαδρομής της λύσης. Για κάθε υποδιαδρομή εάν προκύψει καλύτερη λύση όσον αφορά την διανυόμενη απόσταση από την λύση που υπήρχε πριν την διαδικασία του 2-opt τότε κρατείται η καινούρια λύση. Όταν τελειώσουν όλοι οι πιθανοί συνδυασμοί για την διαδικασία του intra-route ξεκινάει η διαδικασία του inter-route. Επαναλαμβάνεται για κάθε πιθανό ζεύγος ακμών μεταξύ όλων των υποδιαδρομών της λύσης. Εάν προκύψει συνολικά καλύτερη λύση όσον αφορά την διανυόμενη απόσταση από την λύση που προέκυψε από την διαδικασία του intra-route τότε κρατείται η καινούρια λύση.

Για τις λύσεις οι οποίες έχουν βγει εκτός χωρητικότητας ακολουθείται μόνο η διαδικασία του inter-route καθώς γίνονται αποδεκτές μόνο λύσεις που ικανοποιούν το κριτήριο της χωρητικότητας. Κάθε υποδιαδρομή της λύσης αλλά και συνολικά η λύση πρέπει να έχει την απαραίτητη χωρητικότητα για να ικανοποιεί την ζήτηση των κόμβων. Όπως και προηγουμένως η διαδικασία του inter-route γίνεται για κάθε πιθανό ζεύγος ακμών μεταξύ όλων των υποδιαδρομών της λύσης.

Βήμα 7:

Υπολογίζεται η αντικειμενική συνάρτηση της νέας λύσης με βάση την εξίσωση (3.4.1). Εάν η νέα λύση είναι μέχρι 50% χειρότερη από την αρχική λύση που μετατρέψαμε την διατηρούμε στο πλήθος λύσεων καθώς στην συνέχεια μετά από κάποιες μετακινήσεις μπορεί να δώσει καλύτερη λύση.

Βήμα 8:

Επανυπολογίζουμε τις φωτεινότητες, ελκυστικότητες και αποστάσεις καθώς πλέον έχουν αλλάξει οι πυγολαμπίδες και συνεχίζουμε ελέγχοντας τις υπόλοιπες διαδρομές. Ο αλγόριθμος τερματίζει όταν τελειώσουν όλα τα iterations. Στο τέλος κάθε iteration υπολογίζεται η καλύτερη λύση.

Βήμα 9:

Όταν τερματίσει ο αλγόριθμος με το πέρας όλων των iterations, από το σύνολο των καλύτερων λύσεων βρίσκουμε την συνολικά καλύτερη λύση. Η σύγκριση μεταξύ των λύσεων γίνεται με βάση την τιμή της αντικειμενικής συνάρτησης.

Ο ψευδοκώδικας του αλγορίθμου συνοπτικά μπορεί να παρουσιαστεί ως εξής:

Begin

Αντικειμενική συνάρτηση $F(x)$, $x=(x_1, x_2, \dots, x_d)$

Γέννηση αρχικού πληθυσμού n πυγολαμπίδων, x_i , $i=1, 2, \dots, n$

Η φωτεινότητα I_i της x_i πυγολαμπίδας καθορίζεται από την $F(x_i)$

Προσδιόρισε παραμέτρους γ , β_0 , α

While ($t < \text{iterations}$)

 for $i=1 : n$ (για όλες τις πυγολαμπίδες)

 for $j=1 : n$ (για όλες τις πυγολαμπίδες)

 if ($I_j > I_i$)

 μετακίνησε την πυγολαμπίδα i προς την j

 end if

 αναπροσάρμοσε την φωτεινότητα των πυγολαμπίδων μετά την μετακίνηση

 end for j

 end for i

 σύγκρινε τις πυγολαμπίδες και βρες την τωρινή καλύτερη

end while

από το πλήθος των καλύτερων λύσεων βρες την συνολικά καλύτερη

end

Κεφάλαιο 4: Αποτελέσματα

Μετά την ανάπτυξη του αλγορίθμου το επόμενο βήμα είναι να εξαχθούν αριθμητικά αποτελέσματα και να γίνουν κάποια τεστ που να καθορίζουν τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος. Αυτό γίνεται για να διαπιστωθεί το κατά πόσο ο αλγόριθμος είναι έγκυρος, αξιόπιστος και αποτελεσματικός.

Δεδομένου ότι δεν υπάρχουν στην βιβλιογραφία συγκεκριμένα σύνολα δεδομένων (dataset) για το πρόβλημα του HFFVRP χρησιμοποιήθηκαν σύνολα δεδομένων που αναφέρονται στο πρόβλημα του CVRP. Για να γίνει αυτό χρειάστηκαν να γίνουν δύο παραδοχές. Η πρώτη είναι ότι τα σταθερά κόστη θα είναι μηδενικά και η τιμή της αντικειμενικής συνάρτησης θα επηρεάζεται μόνο από τα μεταβλητά κόστη, δηλαδή θα επηρεάζεται από την συνολικά διανυόμενη απόσταση των φορτηγών. Η δεύτερη παραδοχή είναι ότι ο στόλος των οχημάτων θα είναι ομοιογενής, δηλαδή όλα τα φορτηγά θα έχουν ακριβώς την ίδια χωρητικότητα. Με τις παραδοχές αυτές ο αλγόριθμος τροποποιείται λίγο στη υλοποίησή του όμως η γενική εικόνα παραμένει η ίδια. Εάν ανταποκρίνεται σε καλό βαθμό στα σύνολα δεδομένων για το CVRP αυτό σημαίνει ότι ο αλγόριθμος ανταποκρίνεται καλά και για το HFFVRP, δηλαδή είναι αξιόπιστος και αποτελεσματικός.

Τα σύνολα δεδομένων που χρησιμοποιήθηκαν περιλαμβάνουν τον αριθμό των κόμβων, τις συντεταγμένες τους, τον αριθμό των φορτηγών και την χωρητικότητά τους καθώς επίσης και τις βέλτιστες λύσεις που έχουν βρεθεί χρησιμοποιώντας είτε ακριβείς αλγορίθμους είτε ευρετικούς ή μεθευρετικούς αλγορίθμους. Τα συγκεκριμένα σύνολα δεδομένων δημοσιεύθηκαν από τους (Augerat et al., 1995; Christofides and Eilon, 1969). Τα δεδομένα A,B,P αναφέρθηκαν από τους (Augerat et al., 1995), και τα E από τους (Christofides and Eilon, 1969).

Στον πίνακα 4.1 φαίνονται τα σύνολα δεδομένων καθώς και η βέλτιστη λύση του καθενός. Με n συμβολίζεται ο αριθμός των κόμβων ενώ με k ο αριθμός των φορτηγών.

Πίνακας 4.1: Σύνολα δεδομένων

dataset	βέλτιστη λύση
A-n32-k5	784
A-n33-k5	661
A-n33-k6	742
A-n34-k5	778
A-n36-k5	799
A-n37-k5	669
A-n37-k6	949

A-n38-k5	730
A-n39-k5	822
A-n39-k6	831
A-n44-k6	937
A-n45-k6	944
A-n45-k7	1146
A-n46-k7	914
A-n48-k7	1073
A-n53-k7	1010
A-n54-k7	1167
A-n55-k9	1073
A-n60-k9	1354
A-n61-k9	1034
A-n62-k8	1288
A-n63-k9	1616
A-n63-k10	1314
A-n64-k9	1401
A-n65-k9	1174
A-n69-k9	1159
A-n80-k10	1763
B-n31-k5	672
B-n34-k5	788
B-n35-k5	955
B-n38-k6	805
B-n39-k5	549
B-n41-k6	829
B-n43-k6	742
B-n44-k7	909
B-n45-k5	751
B-n45-k6	678
B-n50-k7	741
B-n50-k8	1312
B-n51-k7	1032
B-n52-k7	747
B-n56-k7	707
B-n57-k7	1153
B-n57-k9	1598
B-n63-k10	1496
B-n64-k9	861
B-n66-k9	1316
B-n67-k10	1032
B-n68-k9	1272
B-n78-k10	1221
P-n16-k8	450
P-n19-k2	212

P-n20-k2	216
P-n21-k2	211
P-n22-k2	216
P-n22-k8	603
P-n23-k8	529
P-n40-k5	458
P-n45-k5	510
P-n50-k7	554
P-n50-k8	631
P-n50-k10	696
P-n51-k10	741
P-n55-k7	568
P-n55-k8	588
P-n55-k10	694
P-n60-k10	744
P-n60-k15	968
P-n65-k10	792
P-n70-k10	827
P-n76-k4	593
P-n76-k5	627
P-n101-k4	681
E-n22-k4	375
E-n23-k3	569
E-n30-k3	534
E-n33-k4	835
E-n51-k5	521
E-n76-k7	682
E-n76-k10	830
E-n76-k14	1021

Προτού αναφερθούμε στα αποτελέσματα που προέκυψαν χρησιμοποιώντας τα ανωτέρω σύνολα δεδομένων πρώτα πρέπει να αναφερθούμε σε ένα τεστ που έγινε για να εξακριβώσουμε τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος.

Σύμφωνα με την εξίσωση (3.3.7), που αφορά την μετακίνηση μιας πυγολαμπίδας προς μια άλλη, εμπεριέχεται μια γεννήτρια τυχαίων αριθμών καθώς και ένας παράγοντας τυχειότητας. Για τον λόγο αυτό εάν η μετακίνηση δεν είναι κατευθυνόμενη προς ένα τοπικό ή ολικό βέλτιστο τότε η μετακίνηση γίνεται προς τυχαία κατεύθυνση γίνεται δηλαδή ένας τυχαίος περίπατος. Βλέποντας τις μέσες τιμές της αντικειμενικής συνάρτησης σε κάθε επανάληψη του αλγορίθμου μπορούμε να διαπιστώσουμε κατά πόσο γίνεται τυχαίος περίπατος ή όχι. Αν οι μέσες τιμές δεν έχουν μια τάση να μειώνονται, αλλά έχουν αυξομειώσεις τότε ο αλγόριθμος κινείται με την λογική του τυχαίου περιπάτου. Στην προκειμένη περίπτωση συμβαίνει αυτό ακριβώς το γεγονός. Οι λόγοι για τους οποίους γίνεται αυτό θα εξηγηθούν στο

επόμενο κεφάλαιο που περικλείει τα συμπεράσματα που αφορούν τον αλγόριθμο. Βέβαια πρέπει στο σημείο αυτό να τονιστεί ότι παρά το γεγονός της εμφάνισης του τυχαίου περιπάτου ο αλγόριθμος δεν παύει να είναι έγκυρος ή αποτελεσματικός απλά αυξάνει σε μεγάλο βαθμό η τυχαιότητα με την οποία βρίσκονται οι λύσεις, γίνεται με λίγα λόγια λιγότερο αποτελεσματικός.

Τα συγκεκριμένα σύνολα δεδομένων μπορούν να διαχωριστούν σε δύο κατηγορίες. Η πρώτη αφορά αυτά που είναι σχετικά μικρά από άποψη κόμβων, δηλαδή που έχουν μέχρι 39 κόμβους. Η δεύτερη κατηγορία είναι αυτή που έχει περισσότερους από 40 κόμβους. Ο λόγος που γίνεται αυτός ο διαχωρισμός είναι ότι αναμένεται ο αλγόριθμος να έχει καλύτερη απόδοση στα μικρότερα σύνολα δεδομένων από ότι στα μεγαλύτερα. Αυτό οφείλεται στον γεγονός ότι η μετακίνηση των πυγολαμπίδων δεν είναι κατευθυνόμενη αλλά ακολουθεί την λογική του τυχαίου περιπάτου (random walk). Έτσι στα μικρότερα σύνολα δεδομένων οι επιλογές τις οποίες έχει η εκάστοτε πυγολαμπίδα κατά την μετακίνησή της είναι λιγότερες από ότι στα μεγαλύτερα σύνολα και επομένως μπορεί να βγάλει καλύτερα αποτελέσματα. Δηλαδή είναι μικρότερο το εύρος της τυχαιότητας οπότε τα αποτελέσματα μπορεί να είναι κατά πολύ καλύτερα από αυτά στα μεγάλα σύνολα. Το γεγονός αυτό αποδεικνύεται και από τα αποτελέσματα που προέκυψαν και θα φανούν στους παρακάτω πίνακες.

Στους πίνακες που ακολουθούν παρουσιάζονται τα σύνολα δεδομένων η καλύτερη γνωστή έως τώρα λύση που έχει βρεθεί από την βιβλιογραφία, η καλύτερη λύση που προέκυψε από τον αλγόριθμο που αναπτύχθηκε στην εν λόγω εργασία, η μέση τιμή των λύσεων που βρέθηκαν, καθώς και η ποσοστιαία διαφορά μεταξύ της βέλτιστης λύσης και της λύσης του αλγορίθμου μας. Επίσης παρουσιάζονται και αποτελέσματα που έχουν προκύψει για τα σύνολα δεδομένων από άλλους αλγορίθμους που έχουν αναπτυχθεί και η ποσοστιαία διαφορά τους από τις βέλτιστες λύσεις. Τα αποτελέσματα των άλλων αλγορίθμων που παρουσιάζονται είναι η καλύτερη λύση που βρήκε ο εκάστοτε αλγόριθμος για κάθε συγκεκριμένο σύνολο δεδομένων. Αυτό γίνεται για να υπάρχει σύγκριση των αποτελεσμάτων του αλγορίθμου που αναπτύξαμε με άλλους που υπάρχουν στην βιβλιογραφία. Για κάθε σύνολο δεδομένων ο αλγόριθμος έτρεξε τριάντα φορές δίνοντας τριάντα λύσεις. Στους πίνακες παρουσιάζεται η καλύτερη από αυτές τις 30 λύσεις. Όπως αναφέρεται και στο προηγούμενο κεφάλαιο ο αλγόριθμος έτρεξε χρησιμοποιώντας είκοσι πυγολαμπίδες και για χίλια iterations. Για κάθε σύνολο δεδομένων υπάρχει ένας πίνακας που συμπεριλαμβάνει τα μικρά σύνολα, μέχρι 39 κόμβους, και τα μεγαλύτερα σύνολα με περισσότερους από 40 κόμβους.

Στους πίνακες 4.2 και 4.3 παρουσιάζονται τα αποτελέσματα για τα σύνολα δεδομένων της κατηγορίας A. Ο αλγόριθμος μας συγκρίνεται με έναν υβριδικό ευρετικό αλγόριθμο (SA+NN) που αναπτύχθηκε από τους (Abdelazziz et al., 2014), έναν ευρετικό αλγόριθμο (CLOVES) που αναπτύχθηκε από τους (Ganesh and Narendran, 2007), έναν υβριδικό αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων που αναπτύχθηκε από τους (Chen et al., 2006), έναν εξελικτικό αλγόριθμο (DE) που αναπτύχθηκε από τους (Kromer et al., 2013), έναν υβριδικό αλγόριθμο (SR-GCWS)

που αναπτύχθηκε από τους (Juan et al., 2010) και τέλος έναν αλγόριθμο σάρωσης (SWEEP) που αναπτύχθηκε από τους (Na et al., 2011). Ο πίνακας 4.2 αφορά σύνολα δεδομένων μέχρι 39 κόμβους ενώ ο πίνακας 4.3 σύνολα δεδομένων με περισσότερους από 40 κόμβους

. Πίνακας 4.2: dataset A μέχρι 39 κόμβους

dataset	Best known	SA+NN	%	DPSO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
A-n32-k5	784	1012	29%			1028	31%	787	0,04%	810	3%	831	6%	855
A-n33-k5	661	847	28%	661	0%	837	27%	662	0,01%	686	4%	711	8%	720
A-n33-k6	742	919	24%			967	30%	742	0	743	0,01%	783	6%	794
A-n34-k5	778	933	20%			998	28%			785	1%	827	6%	847
A-n36-k5	799	1126	41%			1080	35%			826	3%	865	8%	875
A-n37-k5	669	876	31%			827	24%	672	0,04%	670	0%	669	0%	689
A-n37-k6	949	1180	24%			1174	24%			962	1%	1021	8%	1053
A-n38-k5	730	920	26%			921	26%	733	0,04%	749	3%	787	8%	809
A-n39-k5	822	1147	40%			1069	30%					898	9%	911
A-n39-k6	831	1065	28%			1237	49%	833	0,02%	856	3%	868	4%	896

Από τον πίνακα 4.2 παρατηρούμε ότι ο αλγόριθμος λειτουργεί εν μέρει ικανοποιητικά στα μικρά σύνολα δεδομένων καθώς οι διαφορές από τις βέλτιστες τιμές κυμαίνονται από 0% έως 9%. Ο αλγόριθμος λειτουργεί πολύ καλύτερα από τους αλγορίθμους (SA+NN) και (DE) καθώς οι αποκλίσεις από τις βέλτιστες τιμές του αλγορίθμου της πυγολαμπίδας είναι κατά πολύ καλύτερες από τις αντίστοιχες των δύο άλλων αλγορίθμων. Επίσης παρατηρούμε ότι ο αλγόριθμος (SWEEP) λειτουργεί καλύτερα από τον δικό μας. Σε σχέση με τους αλγορίθμους (DPSO-SA) και (SR-GCWS) δεν μπορούν να εξαχθούν ασφαλή συμπεράσματα γιατί έχουν αποτελέσματα μόνο για λίγα σύνολα δεδομένων στα οποία βρίσκουν σχεδόν τις βέλτιστες τιμές. Επιπλέον παρατηρούμε ότι η τιμή της μέσης τιμής της αντικειμενικής συνάρτησης όλων των επαναλήψεων όλων των συνόλων έχει μικρή απόκλιση από την βέλτιστη τιμή. Αυτό σημαίνει ότι γενικά όλες οι λύσεις που παίρνουμε από κάθε επανάληψη είναι αρκετά κοντά στην βέλτιστη λύση που βρήκαμε.

Πίνακας 4.3: dataset A με περισσότερους από 40 κόμβους

dataset	Best known	SA+NN	%	DPSO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
A-n44-k6	937	1356	45%			1136	21%			957	2%	1051	12%	1074
A-n45-k6	944	1210	28%			1307	38%	945	0,01%	991	5%	1108	17%	1150
A-n45-k7	1146	1361	19%			1523	33%	1146	0	1173	2%	1281	12%	1301
A-n46-k7	914	1071	17%	928	2%	1316	44%			946	4%	1049	15%	1093
A-n48-k7	1073	1292	20%	1101	3%	1327	24%			1113	4%	1218	14%	1299
A-n53-k7	1010	1261	25%			1512	50%					1200	19%	1238
A-n54-k7	1167	1414	21%			1601	37%					1374	18%	1398
A-n55-k9	1073	1317	23%			1873	75%	1074	0,01%	1095	2%	1324	23%	1354
A-n60-k9	1354	1733	28%	1360	0,1%	2077	53%	1355	0,01%	1420	5%	1650	22%	1670
A-n61-k9	1034	1285	24%			1491	44%	1039	0%	1100	6%	1362	32%	1381
A-n62-k8	1288	1604	25%	1288	0%	2025	57%			1359	6%	1566	22%	1596
A-n63-k9	1616	2001	24%			2324	44%	1622	0%	1712	6%	1976	22%	1995
A-n63-k10	1314	1542	17%			2052	56%			1386	5%	1606	22%	1643
A-n64-k9	1401	1821	30%			2083	49%			1499	7%	1684	20%	1706
A-n65-k9	1174	1429	22%			1980	69%	1181	1%	1223	4%	1522	30%	1583
A-n69-k9	1159	1333	15%			1869	61%			1207	4%	1518	31%	1548
A-n80-k10	1763	2318	31%			2772	57%	1766	0%	1866	6%	2235	27%	2279

Στον πίνακα 4.3 που είναι τα μεγάλα σύνολα δεδομένων παρατηρούμε ότι ο αλγόριθμος δεν λειτουργεί και τόσο αποδοτικά καθώς οι διαφορές κυμαίνονται από 12% έως 32%. Οι ποσοστιαίες διαφορές στους δύο ανωτέρω πίνακες καταδεικνύουν το γεγονός ότι στα μεγαλύτερα σύνολα η τυχαιότητα είναι μεγαλύτερη, δηλαδή κατά την μετακίνηση υπάρχουν πολύ περισσότερες εναλλακτικές στις οποίες μπορεί να κινηθεί η πυγολαμπίδα και επομένως η απόκλιση από την βέλτιστη λύση μπορεί να είναι μεγαλύτερη. Παρ' όλα αυτά ο αλγόριθμός μας σε σχέση με τους (SA+NN) και (DE) εξακολουθεί σχεδόν σε όλα τα σύνολα να παρουσιάζει καλύτερα αποτελέσματα.

Επίσης ο αλγόριθμος (SWEEP) εξακολουθεί να έχει πολύ καλύτερα αποτελέσματα και στα μεγάλα σύνολα δεδομένων ενώ και πάλι δεν μπορούν να εξαχθούν ασφαλή συμπεράσματα για τους άλλους δύο αλγορίθμους καθώς παρουσιάζουν αποτελέσματα μόνο για τα σύνολα στα οποία βρίσκουν την βέλτιστη τιμή ή είναι πολύ κοντά σε αυτήν. Επιπλέον οι μέσες τιμές δεν έχουν μεγάλη διαφορά κατά κύριο λόγο από την καλύτερη λύση που βρήκε ο αλγόριθμος σε κάθε σύνολο όπως συνέβη δηλαδή και στα μικρότερα σύνολα δεδομένων.

Στους πίνακες 4.4 και 4.5 παρουσιάζονται τα αποτελέσματα για τα σύνολα δεδομένων της κατηγορίας B. Ο αλγόριθμος συγκρίνεται με αλγορίθμους που χρησιμοποιήθηκαν και στην σύγκριση της κατηγορίας A. Τέτοιοι είναι οι αλγόριθμοι (SA+NN) και (SWEEP) καθώς και οι αλγόριθμοι (DPSO-SA) και (SR-GCWS) που δεν έχουν όμως αποτελέσματα για όλα τα σύνολα δεδομένων της κατηγορίας B. Ο πίνακας 4.4 αφορά σύνολα δεδομένων μέχρι 39 κόμβους και ο πίνακας 4.5 σύνολα δεδομένων με περισσότερους από 40 κόμβους.

Πίνακας 4.4: dataset B μέχρι 39 κόμβους

dataset	Best known	SA+NN	%	DSPO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
B-n31-k5	672	713	6%					676	1%	677	1%	672	0%	686
B-n34-k5	788	995	26%							802	2%	788	0%	793
B-n35-k5	955	1006	5%	955	0%			956	0%	962	1%	986	3%	1000
B-n38-k6	805	888	10%							817	1%	834	4%	862
B-n39-k5	549	683	24%					553	1%	575	5%	605	10%	621

Από τον πίνακα 4.4 παρατηρούμε ότι ο αλγόριθμος λειτουργεί πάρα πολύ καλά για τα μικρά σύνολα έχοντας μάλιστα βρει την βέλτιστη λύση για το σύνολο B-n31-k5 και για το σύνολο B-n34-k5. Παρατηρούμε ότι είναι πολύ καλύτερος σε σχέση με τον αλγόριθμο (SA+NN) καθώς σε όλα τα σύνολα έχει πολύ καλύτερα αποτελέσματα. Επίσης ο (SWEEP) έχει καλύτερα αποτελέσματα, στα σύνολα που δεν βρήκαμε τις βέλτιστες λύσεις, από τον δικό μας αλγόριθμο αλλά οι ποσοστιαίες διαφορές των δύο αλγορίθμων είναι πολύ μικρότερες από ότι γινόταν στα μικρά σύνολα του dataset A. Επίσης οι αλγόριθμοι (DPSO-SA) και (SR-GCWS) λειτουργούν καλύτερα από τον δικό μας στα σύνολα που εμφανίζουν αποτελέσματα. Επιπλέον για κάθε σύνολο η μέση τιμή όλων των επαναλήψεων έχει μικρή διαφορά από την καλύτερη λύση που βρέθηκε. Με βάση τα ανωτέρω ο αλγόριθμος λειτουργεί πολύ ικανοποιητικά στα μικρά σύνολα.

Πίνακας 4.5: dataset B με περισσότερους από 40 κόμβους

dataset	Best known	SA+NN	%	DPSO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
B-n41-k6	829	927	12%					834	1%	843	2%	919	11%	927
B-n43-k6	742	789	6%							746	1%	793	7%	820
B-n44-k7	909	1212	33%							942	4%	1003	10%	1021
B-n45-k5	751	943	26%	751	0%			754	0,01%	797	6%	843	12%	852
B-n45-k6	678	844	24%							732	8%	751	11%	781
B-n50-k7	741	948	28%					744	0,01%	779	5%	838	13%	864
B-n50-k8	1312	1560	19%							1349	3%	1414	8%	1433
B-n51-k7	1032	1177	14%									1230	19%	1264
B-n52-k7	747	891	19%					749	0,01%	758	1%	889	19%	910
B-n56-k7	707	836	18%					712	1%	726	3%	825	17%	850
B-n57-k7	1153	1477	28%									1469	27%	1492
B-n57-k9	1598	1744	9%					1602	0%	1642	3%	1802	13%	1826
B-n63-k10	1496	1688	13%									1811	21%	1840
B-n64-k9	861	1027	19%					868	1%	1161	35%	1162	35%	1200
B-n66-k9	1316	1578	20%							1363	4%	1560	19%	1587
B-n67-k10	1032	1287	25%					1039	1%	1080	5%	1318	28%	1345
B-n68-k9	1272	1399	10%	1272	0%			1276	0%	1308	3%	1512	19%	1528
B-n78-k10	1221	1363	12%	1239	1%			1228	1%	1268	4%	1625	33%	1642

Από την άλλη στον πίνακα 4.5 παρατηρούμε ότι όσο αυξάνεται ο αριθμός των κόμβων αυξάνεται σε γενικές γραμμές και η απόκλιση από την βέλτιστη λύση. Παρόλα αυτά στο σύνολό τους οι αποκλίσεις για το dataset B στα μεγάλα σύνολα είναι μικρότερες σε γενικές γραμμές από τις αντίστοιχες για το dataset A. Συγκριτικά με τον (SA+NN) ο αλγόριθμός μας είναι πολύ καλύτερος αλλά εξακολουθεί να είναι λιγότερο αποδοτικός από τον αλγόριθμο (SWEEP). Επίσης και πάλι οι άλλοι δύο

αλγόριθμοι παρουσιάζουν αποτελέσματα μόνο σε σύνολα στα οποία είτε έχουν βρει τις βέλτιστες τιμές είτε τιμές με μικρές αποκλίσεις. Στα συγκεκριμένα σύνολα σαφώς έχουν καλύτερα αποτελέσματα από τον δικό μας αλγόριθμο αλλά χωρίς στοιχεία για τα υπόλοιπα σύνολα δεν μπορούμε να εξάγουμε ασφαλή συμπεράσματα για το πώς λειτουργούν συνολικά οι αλγόριθμοι. Όπως συμβαίνει και στα προηγούμενα σύνολα δεδομένων των άλλων πινάκων οι μέσες τιμές έχουν μικρή διαφορά από την λύση που βρήκαμε ως καλύτερη σε κάθε σύνολο.

Στους πίνακες 4.6 και 4.7 παρουσιάζονται τα αποτελέσματα για τα σύνολα δεδομένων της κατηγορίας P. Ο αλγόριθμος συγκρίνεται με αλγορίθμους που χρησιμοποιήθηκαν και στην σύγκριση των προηγούμενων δύο κατηγοριών. Τέτοιοι είναι οι αλγόριθμοι (SA+NN) και (SWEEP) καθώς και ο αλγόριθμος (SR-GCWS) που δεν έχει όμως αποτελέσματα για όλα τα σύνολα δεδομένων της κατηγορίας P. Ο πίνακας 4.6 αφορά σύνολα δεδομένων μέχρι 39 κόμβους και ο πίνακας 4.7 σύνολα δεδομένων με περισσότερους από 40 κόμβους.

Πίνακας 4.6: dataset P μέχρι 39 κόμβους

dataset	Best known	SA+NN	%	DPSO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
P-n16-k8	450	546	21%							513	14%	450	0%	450
P-n19-k2	212	253	19%					212	0%	219	3%	212	0%	212
P-n20-k2	216	267	24%					217	0,01%	217	0%	216	0%	216
P-n21-k2	211	288	36%							211	0	211	0%	211
P-n22-k2	216	274	27%					218	0,01%	216	0	216	0%	217
P-n22-k8	603	667	11%					589	-2%	560	-7%	603	0%	603
P-n23-k8	529	743	40%							554	5%	529	0%	532

Από τον πίνακα 4.6 παρατηρούμε ότι ο αλγόριθμος λειτουργεί πάρα πολύ αποδοτικά για το συγκεκριμένο dataset όσον αφορά τα μικρά σύνολα δεδομένων. Συγκεκριμένα βρίσκει τις βέλτιστες τιμές για όλα τα μικρά σύνολα δεδομένων και είναι καλύτερος συνολικά από όλους τους άλλους αλγορίθμους με τους οποίους συγκρίνεται. Υστερεί μόνο στο σύνολο P-n22-k8 στο οποίο τόσο ο αλγόριθμος (SR-GCWS) όσο και ο (SWEEP) βρίσκουν καλύτερη τιμή από την βέλτιστη. Επιπλέον σε όλα τα σύνολα εκτός από το τελευταίο οι μέσες τιμές όλων των επαναλήψεων είναι ακριβώς ίδιες με τις βέλτιστες λύσεις που βρέθηκαν. Τα συγκεκριμένα σύνολα αποτελούν την απόδειξη ότι στα μικρά σύνολα δεδομένων όπου η τυχαιότητα λόγω του τυχαίου περιπάτου είναι μικρή ο αλγόριθμός μας είναι πάρα πολύ αποδοτικός.

Πίνακας 4.7: dataset P με περισσότερους από 40 κόμβους

dataset	Best known	SA+NN	%	DPSO-SA	%	DE	%	SR-GCWS	%	SWEEP	%	firefly	%	mean
P-n40-k5	458	563	23%					461	0,01%	467	2%	508	11%	528
P-n45-k5	510	662	30%									595	17%	603
P-n50-k7	554	647	17%									689	24%	704
P-n50-k8	631	721	14%					634	0%			828	31%	843
P-n50-k10	696	808	16%					700	1%			843	21%	859
P-n51-k10	741	857	16%					741	0%			895	21%	904
P-n55-k7	568	679	20%									670	18%	682
P-n55-k8	588	690	17%									741	26%	750
P-n55-k10	694	832	20%									853	23%	861
P-n60-k10	744	896	20%					748	1%			940	26%	949
P-n60-k15	968	1159	20%									1195	23%	1205
P-n65-k10	792	964	22%					796	1%			1020	29%	1037
P-n70-k10	827	989	20%					830	0%			1119	35%	1131
P-n76-k4	593	753	27%	602	2%			598	1%	612	3%	733	24%	741
P-n76-k5	627	671	7%					633	1%			832	33%	840
P-n101-k4	681	891	31%	694	2%			692	2%	715	5%	874	28%	887

Από τον πίνακα 4.7 παρατηρούμε ότι στα μεγαλύτερα σύνολα δεδομένων της συγκεκριμένης κατηγορίας οι αποκλίσεις είναι μεγάλες όπως και στα αντίστοιχα σύνολα από τα dataset A και B. Παρότι ο αλγόριθμος είναι πολύ αποδοτικός για τα μικρότερα σύνολα δεδομένων αυτής της κατηγορίας στα μεγαλύτερα σύνολα εμφανίζει μεγάλες αποκλίσεις. Αυτό οφείλεται όπως και στα άλλα σύνολα δεδομένων στην μεγαλύτερη τυχαιότητα που υπάρχει λόγω του τυχαίου περιπάτου. Συγκριτικά με τον αλγόριθμο (SA+NN) ο δικός μας αλγόριθμος σε μερικά σύνολα παρουσιάζει καλύτερα αποτελέσματα ενώ σε κάποια άλλα χειρότερα. Όλοι οι υπόλοιποι αλγόριθμοι με τους οποίους γίνονται συγκρίσεις παρουσιάζουν αποτελέσματα μόνο

σε σύνολα στα οποία βρίσκουν την βέλτιστη τιμή ή έχουν μικρές αποκλίσεις από την βέλτιστη λύση. Αξίζει να σημειωθεί ότι και εδώ οι αποκλίσεις των μέσων τιμών των επαναλήψεων από τις βέλτιστες λύσεις που βρέθηκαν είναι πολύ μικρές.

Στους πίνακες 4.8 και 4.9 παρουσιάζονται τα αποτελέσματα για τα σύνολα δεδομένων της κατηγορίας E. Ο αλγόριθμος συγκρίνεται με αλγορίθμους που χρησιμοποιήθηκαν και στην σύγκριση των συνόλων δεδομένων A,B και P. Τέτοιοι είναι οι αλγόριθμοι (CLOVES), (SWEEP), (DPSO-SA) και (SR-GCWS). Ο πίνακας 4.8 αφορά σύνολα δεδομένων μέχρι 39 κόμβους και ο πίνακας 4.9 σύνολα δεδομένων με περισσότερους από 40 κόμβους.

Πίνακας 4.8: dataset E μέχρι 39 κόμβους

dataset	Best known	CLOVES	%	SR-GCWS	%	DPSO-SA	%	SWEEP	%	firefly	%	mean
E-n22-k4	375	375	0%	375	0%			375	0%	375	0%	375
E-n23-k3	569	569	0%					569	0%	569	0%	569
E-n30-k3	534	534	0%	505	-5%	534	0%	543	2%	534	0%	537
E-n33-k4	835	835	0%	837	0,01%			852	2%	835	0%	842

Από τον πίνακα 4.8 παρατηρούμε ότι ο αλγόριθμος είναι ιδιαίτερα αποδοτικός στα μικρά σύνολα της συγκεκριμένης κατηγορίας. Και για τα τέσσερα σύνολα βρήκε τις βέλτιστες τιμές καθώς και οι μέσες τιμές στο εκάστοτε σύνολο δεδομένων είτε είναι ίδιες με την βέλτιστη τιμή είτε έχουν πάρα πολύ μικρή απόκλιση. Επιβεβαιώνεται δηλαδή και πάλι ότι για τα μικρά σύνολα δεδομένων ο αλγόριθμος λειτουργεί αποδοτικά και αποτελεσματικά. Σε σχέση με τους άλλους αλγορίθμους με τους οποίους συγκρίνεται δεν υπάρχει κάποια ουσιαστική διαφορά. Όλοι λειτουργούν πολύ καλά για τα συγκεκριμένα σύνολα.

Πίνακας 4.9: dataset E με περισσότερους από 40 κόμβους

dataset	Best known	CLOVES	%	SR-GCWS	%	DPSO-SA	%	SWEEP	%	firefly	%	mean
E-n51-k5	521	521	0%	525	1%	528	1%	532	2%	626	20%	643
E-n76-k7	682	690	1%	689	1%	688	1%	703	3%	922	35%	965
E-n76-k10	830	867	4%	835	1%			907	9%	1186	43%	1203
E-n76-k14	1021	1032	1%	1024	0,01%			1072	5%	1406	38%	1429

Από την άλλη πλευρά στον πίνακα 4.9 βλέπουμε ότι οι αποκλίσεις είναι αρκετά μεγάλες στα μεγαλύτερα σύνολα επιβεβαιώνοντας έτσι ότι στα μεγάλα σύνολα ο αλγόριθμος δεν λειτουργεί τόσο καλά όσο στα μικρότερα σύνολα. Το ίδιο ισχύει και για τις αποκλίσεις των μέσων τιμών που είναι μεγαλύτερες από αυτές που υπάρχουν στα μικρότερα σύνολα. Οι υπόλοιποι τέσσερις αλγόριθμοι με τους οποίους γίνονται συγκρίσεις λειτουργούν πολύ καλύτερα στα συγκεκριμένα σύνολα από τον δικό μας αλγόριθμο.

Συνοψίζοντας τα αποτελέσματα του αλγορίθμου μπορούμε να πούμε ότι στα μικρά σύνολα δεδομένων όλων των κατηγοριών ο αλγόριθμος είναι πάρα πολύ αποδοτικός και αποτελεσματικός. Γενικά είτε βρίσκει τις βέλτιστες τιμές είτε οι αποκλίσεις από τις βέλτιστες τιμές είναι ποσοστιαία μικρές. Επιπλέον οι μέσες τιμές των επαναλήψεων για όλα τα σύνολα δεδομένων είναι πολύ κοντά στην εκάστοτε καλύτερη λύση που βρέθηκε από τον αλγόριθμό μας. Συγκεκριμένα σε δεκατέσσερις περιπτώσεις βρέθηκαν οι βέλτιστες λύσεις στα σύνολα αυτά. Τα σύνολα στα οποία βρέθηκαν οι βέλτιστες λύσεις είναι τα εξής:

A-n37-k5, B-n31-k5, B-n34-k5, P-n16-k8, P-n19-k2, P-n20-k2, P-n21-k2, P-n22-k2, P-n22-k8, P-n23-k8, E-n22-k4, E-n23-k3, E-n30-k3, E-n33-k4

Αντιθέτως όσον αφορά τα μεγάλα σύνολα δεδομένων σε όλα παρατηρούνται μεγαλύτερες αποκλίσεις από αυτές που εμφανίζονται στα μικρότερα σύνολα. Δηλαδή ο αλγόριθμος της πυγολαμπίδας δεν είναι το ίδιο αποδοτικός στα σύνολα αυτά. Παρ' όλα αυτά στις κατηγορίες A,B και P ο αλγόριθμος έχει καλύτερα αποτελέσματα από τον αλγόριθμο (SA+NN) με τον οποίο γίνονται συγκρίσεις και προκύπτουν ασφαλή συμπεράσματα καθώς οι υπόλοιποι αλγόριθμοι έχουν αποσπασματικά αποτελέσματα. Στην κατηγορία όμως E όλοι οι άλλοι αλγόριθμοι εμφανίζουν καλύτερα αποτελέσματα. Οι μέσες τιμές στα μεγαλύτερα σύνολα έχουν επίσης μεγαλύτερη απόκλιση από την καλύτερη λύση που βρέθηκε από τον αλγόριθμό μας σε κάθε σύνολο.

Κεφάλαιο 5: Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή ασχοληθήκαμε με το πρόβλημα της δρομολόγησης ετερογενούς στόλου οχημάτων με σταθερό αριθμό (heterogeneous fixed fleet vehicle routing problem - HFFVRP). Για την επίλυσή του αναπτύχθηκε ένας αλγόριθμος πυγολαμπίδας (firefly algorithm) με την χρήση μεθόδου τοπικής αναζήτησης (2-opt). Για τον έλεγχο της αποδοτικότητας και αποτελεσματικότητάς του χρησιμοποιήθηκε στην επίλυση γνωστών συνόλων δεδομένων που εμπίπτουν στην κατηγορία του προβλήματος δρομολόγησης οχημάτων με συγκεκριμένο όριο χωρητικότητας (capacitated vehicle routing problem). Ο λόγος που έγινε αυτό είναι επειδή δεν υπάρχουν στην βιβλιογραφία σύνολα δεδομένων για την κατηγορία του

HFFVRP. Στο παρόν κεφάλαιο θα ασχοληθούμε με την αξιολόγηση και συζήτηση των αποτελεσμάτων που προέκυψαν από τον τρόπο λειτουργίας του αλγορίθμου.

Κατ' αρχάς πρέπει να διευκρινιστεί ότι ο αλγόριθμος της πυγολαμπίδας δημιουργήθηκε για την επίλυση συνεχών προβλημάτων. Το πρόβλημα με το οποίο καταπιάνεται η συγκεκριμένη διατριβή είναι διακριτό πρόβλημα επομένως για να μπορεί να εφαρμοστεί ο αλγόριθμος πρέπει να γίνουν κατάλληλες τροποποιήσεις.

Ένα σημαντικό ζήτημα του αλγορίθμου είναι ο χώρος στον οποίο βρίσκονται οι αρχικές λύσεις και στον οποίο γίνεται η μετακίνηση της κάθε πυγολαμπίδας. Στα συνεχή προβλήματα οι αρχικές λύσεις είναι τυχαία κατανεμημένες στον χώρο επίλυσης του προβλήματος ο οποίος είναι οριοθετημένος εξ αρχής. Με την μετακίνηση της κάθε πυγολαμπίδας προς μια άλλη πυγολαμπίδα που είναι περισσότερο φωτεινή μειώνεται η μεταξύ τους απόσταση. Με απλά λόγια οι πυγολαμπίδες έχουν την τάση να συγκλίνουν προς τοπικά ή ολικά βέλτιστα και επομένως η μετακίνηση να γίνεται ουσιαστικά κατευθυνόμενη προς την καλύτερη λύση του προβλήματος. Παρ' όλα αυτά στην εξίσωση που διέπει την μετακίνηση των πυγολαμπίδων υπάρχει και ένας όρος που αναφέρεται στην τυχαιότητα. Έτσι εφόσον καμία πυγολαμπίδα δεν είναι πιο φωτεινή από τις υπόλοιπες η μετακίνηση γίνεται τυχαία, δηλαδή πραγματοποιείται ένας τυχαίος περίπατος. Η βελτιστοποίηση της λύσης προκύπτει από την συνεχή ανανέωση της θέσης κάθε πυγολαμπίδας βασιζόμενη στην φωτεινότητα και την ελκυστικότητα.

Το πρόβλημα το οποίο επιλύεται σε αυτή την διατριβή ανήκει στην κατηγορία των διακριτών προβλημάτων. Παρατηρώντας την διακύμανση της μέσης τιμής της αντικειμενικής συνάρτησης σε κάθε επανάληψη διαπιστώθηκε ότι η μετακίνηση των πυγολαμπίδων γίνεται με βάση τον τυχαίο περίπατο και δεν είναι κατευθυνόμενη. Αυτό οφείλεται στο γεγονός ότι με την κάθε μετακίνηση ο αλγόριθμος δεν διασφαλίζει ότι η απόσταση μεταξύ των πυγολαμπίδων μειώνεται. Αυτό γίνεται γιατί σε κάθε μετακίνηση που γίνεται με βάση την εξίσωση (3.3.7) υπάρχει και ο όρος της απόστασης που υπολογίζεται από την εξίσωση (3.3.6). Πρακτικά σε κάθε μετακίνηση η λύση ανασκευάζεται από την αρχή. Λόγω των πράξεων δηλαδή της εξίσωσης (3.3.6) η θέση στην οποία θα βρεθεί η νέα πυγολαμπίδα είναι τυχαία. Επιπλέον λόγω του γεγονότος ότι η λύση μπορεί να πάρει μόνο διακριτές τιμές άλλες φορές η απόσταση της νέας λύσης από την φωτεινότερη λύση προς την οποία μετακινήθηκε, μετά την αποκωδικοποίηση της λύσης, μπορεί να μειωθεί ενώ άλλες φορές μπορεί και όχι.

Το ίδιο πρόβλημα, δηλαδή η χρήση του τυχαίου περιπάτου, εμφανίστηκε και σε άλλη προσπάθεια που έγινε να επιλυθεί ένα διακριτό πρόβλημα με την χρήση του αλγορίθμου της πυγολαμπίδας. Συγκεκριμένα οι (Jati et al., 2011) έλυσαν το πρόβλημα του περιοδεύοντος πωλητή (TSP) με την χρήση ενός εξελικτικού διακριτού αλγορίθμου πυγολαμπίδας. Για τον υπολογισμό της απόστασης αντί να χρησιμοποιήσουν την ευκλείδεια απόσταση, όπως κάνουμε και εμείς στην παρούσα εργασία, χρησιμοποίησαν την εξίσωση: $r=(A/n) * 10$ όπου το A είναι ο συνολικός

αριθμός των διαφορετικών ακμών μεταξύ δύο πυγολαμπίδων και το n ο αριθμός των κόμβων. Βασιζόμενοι σε αυτό η μετακίνηση της πυγολαμπίδας γίνεται με βάση την ακόλουθη εξίσωση: $x_i = \text{rand}(2, A)$. Η απόφασή τους αυτή να ξεφύγουν από τους τύπους του κλασικού αλγορίθμου της πυγολαμπίδας όσον αφορά την απόσταση και την μετακίνηση δεν βοήθησε καθώς κατά την μετακίνησή της η εκάστοτε πυγολαμπίδα δεν έχει προς τα πού να κινηθεί και επομένως ακολουθεί και αυτή τον τυχαίο περίπατο. Στην προσπάθειά τους να λύσουν το πρόβλημα του τυχαίου περιπάτου δημιούργησαν έναν άλλον τρόπο υπολογισμού της μετακίνησης που περιγράφεται αναλυτικά στους (Jati et al., 2013).

Στο σημείο αυτό σχετικά με το πρόβλημα του τυχαίου περιπάτου πρέπει να τονιστεί ότι στην παρούσα εργασία δεν μπορούμε να ακολουθήσουμε τον τρόπο που παρουσιάζεται από τους (Jati et al., 2013). Αυτό οφείλεται στο γεγονός ότι στο HFFVRP όπως και γενικά στο VRP για την εύρεση μιας λύσης χρησιμοποιούνται περισσότερα του ενός φορτηγά, δηλαδή έχει αρκετά περάσματα από την αποθήκη. Αντιθέτως στο TSP για την λύση του προβλήματος υπάρχει μόνο ένα πέρασμα από την αποθήκη. Επομένως αν προσπαθούσαμε να χρησιμοποιήσουμε τον δικό τους τρόπο επίλυσης δεν θα βγάζαμε σωστά αποτελέσματα καθώς θα υπήρχαν πολλές επικαλύψεις όσον αφορά τον αριθμό των διαφορετικών ακμών που υπάρχουν σε δύο πυγολαμπίδες λόγω του γεγονότος ότι η αποθήκη, κάθε φορά που ένα καινούριο φορτηγό μπαίνει στην λύση ουσιαστικά επιστρέφουμε στην αποθήκη, εμφανίζεται πολλές φορές μέσα σε μία λύση.

Το δεύτερο σημαντικό συμπέρασμα της παρούσας μεταπτυχιακής διατριβής αφορά την αποδοτικότητα και αποτελεσματικότητα του αλγορίθμου που αναπτύχθηκε. Για να δούμε πόσο αποδοτικός είναι τρέξαμε τον αλγόριθμο σε γνωστά σύνολα δεδομένων που αφορούν το πρόβλημα του CVRP και τον συγκρίναμε με άλλους αλγορίθμους επίλυσης αυτού του προβλήματος. Εάν ο αλγόριθμος συμπεριφέρεται καλά στην επίλυση του CVRP τότε υποθέτουμε ότι θα συμπεριφέρεται καλά και στην επίλυση του HFFVRP καθώς οι μόνες διαφορές είναι ότι θα προστεθούν τα σταθερά κόστη για κάθε τύπο οχήματος και ότι κάθε όχημα δεν θα έχει την ίδια χωρητικότητα με όλα τα υπόλοιπα.

Από τα αποτελέσματα που προέκυψαν παρατηρούμε ότι ο αλγόριθμος λειτουργεί πολύ αποδοτικά και αποτελεσματικά για τα μικρά σύνολα δεδομένων, έως 39 κόμβους, και λιγότερο αποδοτικά στα μεγαλύτερα σύνολα. Για την ακρίβεια όλες οι βέλτιστες τιμές που βρέθηκαν για τα διάφορα σύνολα αφορούσαν προβλήματα με λιγότερους από 39 κόμβους. Επιπλέον στα μικρά σύνολα οι διαφορές των λύσεων που προέκυψαν από τον αλγόριθμο όταν δεν ήταν οι βέλτιστες έχουν πολύ μικρές αποκλίσεις από τις βέλτιστες λύσεις του κάθε προβλήματος. Βέλτιστες λύσεις βρέθηκαν σε δεκατέσσερα από τα μικρά σύνολα.

Σε αντίθεση με τα μικρά σύνολα στα μεγάλα σύνολα ο αλγόριθμος δεν φαίνεται να λειτουργεί το ίδιο αποτελεσματικά και αποδοτικά. Για την ακρίβεια σε κανένα από αυτά τα σύνολα ο αλγόριθμος δεν κατάφερε να βρει την βέλτιστη λύση και οι

διαφορές των λύσεων που παρήγαγε από τις βέλτιστες λύσεις είναι σημαντικά μεγαλύτερες από ότι στις αντίστοιχες για τα μικρά σύνολα. Παρ' όλα αυτά ο αλγόριθμος κατάφερε να έχει σε γενικές γραμμές καλύτερη επίδοση στα μεγάλα σύνολα από έναν από τους υπόλοιπους αλγόριθμους με τους οποίους συγκρίνεται.

Ο λόγος για τον οποίο υπάρχουν αυτές οι διαφορές μεταξύ των μικρών και των μεγάλων συνόλων είναι η τυχαιότητα που υπάρχει στην διαδικασία εύρεσης της κάθε λύσης. Οφείλεται δηλαδή στον τυχαίο περίπατο κατά την ανανέωση της θέσης της κάθε πυγολαμπίδας. Στα μικρά σύνολα όπου ο αριθμός των κόμβων και των φορτηγών είναι σχετικά μικρός οι πιθανότητες ο αλγόριθμος, παρ' όλο που χρησιμοποιεί τυχαίο περίπατο, να βρει την βέλτιστη λύση είναι αρκετά μεγαλύτερες από ότι στα μεγάλα σύνολα. Αυτό γίνεται γιατί οι εναλλακτικές που έχει για το που θα κινηθεί και πως θα κατανεμηθούν στην λύση οι κόμβοι και τα φορτηγά είναι σαφώς λιγότερες. Για αυτό άλλωστε όλες οι βέλτιστες λύσεις που βρέθηκαν από τον αλγόριθμο αφορούν τα μικρά σύνολα δεδομένων. Επιπλέον αυτός είναι και ο λόγος για τον οποίο οι διαφορές από τις βέλτιστες λύσεις είναι μικρές στα μικρά σύνολα δεδομένων ενώ στα μεγάλα σύνολα είναι μεγαλύτερες, αφού οι εναλλακτικές είναι αρκετά περισσότερες.

Τέλος αναφορά πρέπει να γίνει και στην μέση τιμή όλων των επαναλήψεων που έγιναν για κάθε σύνολο δεδομένων. Στα μικρά σύνολα η μέση τιμή έχει μικρή απόκλιση από την καλύτερη λύση που βρέθηκε από τον αλγόριθμό μας. Επιπλέον η διαφορά της από τις βέλτιστες λύσεις της βιβλιογραφίας να μεν είναι μεγαλύτερη από την καλύτερη λύση που δίνει ο αλγόριθμος αλλά η διαφορά αυτή είναι σε ανεκτά επίπεδα. Για τα μεγάλα σύνολα δεδομένων επίσης η διαφορά της μέσης τιμής δεν είναι πολύ μεγάλη από την λύση που προκύπτει από τον αλγόριθμο αλλά σε σχέση με τις βέλτιστες λύσεις της βιβλιογραφίας η διαφορά για όλα τα σύνολα είναι σαφώς αρκετά μεγαλύτερη. Το γεγονός αυτό δείχνει ότι παρότι κατά την διαδικασία εύρεσης της λύσης γίνεται τυχαίος περίπατος οι λύσεις που δίνει ο αλγόριθμος είναι σχετικά κοντά η μία στην άλλη δηλαδή δεν έχουν τεράστιες διαφορές. Μεγαλύτερες διαφορές παρουσιάζει από τις βέλτιστες λύσεις της βιβλιογραφίας στα σύνολα που η λύση του αλγορίθμου έχει επίσης μεγάλη απόκλιση από την βέλτιστη λύση. Παρότι δηλαδή η μετακίνηση δεν είναι κατευθυνόμενη το εύρος της κίνησης της κάθε πυγολαμπίδας δεν είναι πολύ μεγάλο αλλά όντως κινείται κοντά στην φωτεινότερη πυγολαμπίδα. Για τον λόγο αυτό σε πολλά σύνολα δεδομένων βρίσκουμε και τις βέλτιστες λύσεις.

5.1 Μελλοντικές προεκτάσεις

Στην σχετική βιβλιογραφία που υπάρχει γενικά για τα προβλήματα δρομολόγησης στο πρόβλημα του HFFVRP έχει δοθεί πολύ μικρότερη προσοχή σε σχέση με άλλες παραλλαγές του γενικού προβλήματος δρομολόγησης. Επιπλέον οι προσπάθειες που έγιναν για την επίλυση παραλλαγών του κλασικού προβλήματος δρομολόγησης με καινούριους μεθευρετικούς αλγορίθμους νοημοσύνης σμήνους είναι ελάχιστες.

Πιστεύουμε ότι οι αλγόριθμοι νοημοσύνης σμήνους και συγκεκριμένα ο αλγόριθμος της πυγολαμπίδας πρέπει να εφαρμοστούν και σε άλλες παραλλαγές για την επίλυσή τους. Ο αλγόριθμος που αναπτύχθηκε είναι πολλά υποσχόμενος όσον αφορά την επίδοσή του στην λύση τέτοιου είδους προβλημάτων παρότι αρχικά δεν είχε αναπτυχθεί για την επίλυση διακριτών προβλημάτων.

Σίγουρα σε μελλοντική έρευνα πρέπει να διερευνηθεί ο τρόπος με τον οποίο θα απαλειφθεί ο τυχαίος περίπατος κατά την ανανέωση της θέσης κάθε πυγολαμπίδας. Επειδή η μετακίνηση δεν είναι κατευθυνόμενη προς τοπικά ή ολικά βέλτιστα η ποιότητα των λύσεων δεν είναι πάντα η βέλτιστη και παρουσιάζονται στα μεγάλα σύνολα δεδομένων μεγάλες αποκλίσεις από την βέλτιστη λύση. Εφόσον εξαλειφθεί ο τυχαίος περίπατος τότε πιστεύουμε ότι η επίδοση του αλγορίθμου θα είναι ακόμα καλύτερη και θα μπορεί να βρίσκει ποιοτικότερες λύσεις και στα μεγάλα σύνολα δεδομένων. Μια μελλοντική έρευνα πρέπει να κινηθεί προς αυτή την κατεύθυνση.

Βιβλιογραφία

Abdelaziz, M.M., El-Ghareeb, H.A., Ksasy, M.S.M. (2014), Hybrid heuristic algorithm for solving capacitated vehicle routing problem, *International journal of computers and technology*, Vol. 12 No.9, pp. 3844-3851

Abdullah, A., Deris, S., Mohamad, M., Hashim, S. (2012), A new hybrid firefly algorithm for complex and nonlinear problem, *Distributed computing and artificial intelligence*, pp. 673-680

Abedinia, O., Amjady, N., Naderi, M. (2012), Multi-objective environmental/economic dispatch using firefly technique, in: The 11th international conference on environment and electrical engineering, (EEEIC), IEEE, pp. 461-466

Abshouri, A., Meybodi, M., Bakhtiary, A. (2011), New firefly algorithm based on multi swarm & learning automata in dynamic environments, in: *IEEE proceedings*, pp. 73-77

Ali, N., Othman, M.A., Husain, M.N., Misran, M.H. (2014), A review of firefly algorithm, *ARPJ journal of engineering and applied sciences*, Vol. 9 No. 10, pp. 1732-1736

Anily, S. (1996), The vehicle routing problem with delivery and back-haul options, *Naval Research Logistics*, Vol. 43, pp. 415-434

Archetti, C., Mansini, R., Speranza, M.G. (2001), The split delivery vehicle routing problem with small capacity, *Technical Report no. 201*, Department of quantitative methods, University of Brescia

Arora, S. and Singh, S. (2013), The firefly optimization algorithm: convergence analysis and parameter selection, *International journal of computer applications*, Vol.69 No.3, pp. 48-52

Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D., Rinaldi, G. (1995), Computational results with a branch and cut code for the capacitated vehicle routing problem, *Research report 949-M*, University Joseph Fourier, Grenoble, France

Baldacci, R., Battarra, M., Vigo, D. (2008), Routing a heterogeneous fleet of vehicles, in: Golden, B., Raghavan, S., Wasil, E., (Eds.), *The vehicle routing problem: Latest advances and new challenges*, Springer, pp. 3-27

Ball, M., Magnanti, T., Monma, C., Nemhauser, G. (1995), Network Routing, *Handbooks in Operations Research and Management Science*, Vol. 8

Ballou, R.H. (1999), *Business Logistics Management: Planning, Organizing and Controlling the Supply Chain 4th international edition*, Prentice-Hall International, Upper Saddle River, NJ

- Blasum, U. and Hochstattler, W. (2000), Application of the branch and cut method to the vehicle routing problem, *Zentrum fur Angewandte informatik Koln*, Technical report zpr 386
- Brandao, J. (2006), A new tabu search for the vehicle routing problem with backhauls, *European Journal of Operational Research*, Vol. 173 No.2, pp. 540-555
- Brandao, J., (2011), A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, *Computers & Operations Research*, Vol. 38 No.1, pp. 140-151
- Chai-ead, N., Aungkulanon, P., Luangpaiboon, P. (2011), Bees and firefly algorithms for noisy non-linear optimization problems, in: Proceedings of the international multi conference of engineering and computer scientists, Vol. 2, pp. 1449-1454
- Chandrasekaran, K., Simon, S. (2012), Network and reliability constrained unit commitment problem using binary real coded firefly algorithm, *International journal of electrical power & energy systems*, Vol. 43 No.1, pp. 921-932
- Chen, S., Golden, B., Wasil, E., (2007), The split delivery vehicle routing problem: applications, algorithms, test problems, and computational results, *Networks*, Vol. 49 No.4, pp. 318-329
- Chen, A., Yang, G., Wu, Z. (2006), Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem, *Journal of Zhejiang university SCIENCE A*, Vol. 7 No.4, pp. 607-614
- Chiang, W.C. and Russell, R.A. (2004), A metaheuristic for the vehicle-routing problem with soft time windows, *Journal Of the Operational Research Society*, Vol. 55, pp. 1298-1310
- Choi, E., Tcha, D.W. (2007), A column generation approach to the heterogeneous fleet vehicle problem, *Computers & Operations Research*, Vol. 34 No.7 , pp. 2080-2095
- Christofides, N. and Eilon, S. (1969), An algorithm for the vehicle dispatching problem, *Operational research quarterly*, Vol.20 No.3, pp. 309-318
- Christofides, N., Mingozzi, A., Toth, P. (1979), The vehicle routing problem, in: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), *Combinatorial optimization*, pp. 325-338, Chichester, Willey
- Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002), VRP with time windows, in: Toth, P and Vigo, D. (eds.), *The vehicle routing problem*, SIAM Monographs on discrete mathematics and applications, Vol. 9, pp. 157-193, Philadelphia, PA

- Cordeau, J.F., Laporte, G., Mercier, A. (2001), A unified tabu search heuristic for vehicle routing problems with time windows, *Journal of the Operational Research Society*, Vol. 52, pp. 928-936
- Cordeau, J.F., Laporte, G., Ropke, S. (2008), Recent models and algorithms for one-to-one pickup and delivery problems, In: Golden, B.I., Raghavan, S., Wasil, E.A. (Eds.), *Vehicle Routing: Latest advances and challenges*, pp. 327-357, Springer
- Crainic, T.G., Laporte, G., (1998), *Fleet management and logistics*, Kluwer Academic, London, Dordrecht, Boston
- Crevier, B., Cordeau, J-F., Laporte, G. (2007), The multi-depot vehicle routing problem with inter-depot routes, *European Journal of Operational Research*, Vol. 176 No.2 , pp. 756-773
- Dantzig, G.B. and Ramser, J.M. (1959), The truck dispatching problem, *Management Science*, Vol. 6 , pp. 81-91
- Dell'Amico, M., Righini, G., Salani, M. (2006), A branch-and-price algorithm for the vehicle routing problem with simultaneous pick-up and delivery, *Transportation Science*, Vol. 40, pp. 235-247
- Dror, M., Laporte, G., Trudeau, P. (1994), Vehicle routing with split deliveries, *Discrete Applied Mathematics*, Vol. 50 No.3, pp. 239-254
- Dueck, G. (1993), New optimization heuristics: the great deluge algorithm and the record-to-record travel, *Journal of computational physics*, Vol. 104, pp. 86-92
- Dumas, Y., Desrosiers, J., Soumis, F. (1991), The pickup and delivery problem with time windows, *European Journal of Operational Research*, Vol. 54, pp. 7-22
- Falcon, R., Almeida, M., Nayak, A. (2011), Fault identification with binary adaptive fireflies in parallel and distributed systems, in: IEEE congress on evolutionary computation (CEC), IEEE, pp. 1359-1366
- Farahani, S.M., Abshouri, A., Nasiri, B., Meybodi, A. (2011), A Gaussian firefly algorithm, *International journal of machine learning and computing*, Vol.1 No.5, pp. 448-454
- Farahani, S., Nasiri, B., Meybodi, M. (2011), A multiswarm based firefly algorithm in dynamic environments, in: The third international conference on signal processing systems, (ICSPS), Vol. 3, pp. 68-72
- Farahani, S., Abshouri, A., Nasiri, B., Meybodi, M. (2012), Some hybrid models to improve firefly algorithm performance, *International journal of artificial intelligence*, Vol.8 No.12, pp. 97-117

- Fisher, M.L. (1988), Optimal solution of vehicle routing problems using minimum k-Trees, *Operations research*, Vol. 42, pp. 626-642
- Fister Jr., I., Yang, X.S., Fister, I., Brest, J. (2012), Memetic firefly algorithm for combinatorial optimization, in: Filipic, B., Silc, J. (eds.), *Bioinspired optimization methods and their applications: Proceedings of the fifth international conference on bioinspired optimization methods and their applications*, (BIOMA 2012), Jozef Stefan institute, pp. 75-86
- Fister, I., Fister Jr., I., Yang, X.S., Brest, J. (2013), A comprehensive review of firefly algorithms, *Swarm and evolutionary computation*, Vol. 13, pp. 34-46
- Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragao, M., Reis, M., Uchoa, E., Werneck, R.F. (2006), Robust branch-and-cut-and-price for the capacitated vehicle routing problem, *Mathematical Programming, Series A* 106, pp. 491-511
- Gandomi, A.H., Yang, X.S., Alavi, A.H. (2011), Mixed variable structural optimization using firefly algorithm, *Computers & Structures*, Vol. 89, pp. 2325-2336
- Ganesh, K. and Narendran, T.T. (2007), CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up, *European journal of operational Research*, Vol. 178, pp. 699-717
- Gendreau, M., Laporte, G., Seguin, R. (1996), Stochastic vehicle routing, *European Journal of Operational Research*, Vol. 88, pp. 3-12
- Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D., (1999), A tabu search heuristic for the heterogeneous fleet vehicle routing problem, *Computers & Operations Research*, Vol. 26, pp. 1153-1173
- Glover, F., Laguna, M. (1997), *Tabu search*, Kluwer, Boston
- Goetschalckx, M. and Jacobs-Blecha, C. (1988), The Vehicle Routing Problem with Backhauls: An Effective Heuristic Based on the Generalized Assignment Problem with Ring Metric Norm, *Material Handling Research Center Technical Report* TR-88-13.
- Goetschalckx, M. and Jacobs-Blecha, C. (1989), The vehicle routing problem with backhauls, *European Journal of Operational Research*, Vol. 42 No.1, pp. 39-51
- Goetschalckx, M. and Jacobs-Blecha, C. (1993), The Vehicle Routing Problem with Backhauls: An Optimization Based Approach *.Proceedings of the 2nd Industrial Engineering Research Conference*, May 26-27, Los Angeles, California, pp. 504-509.
- Golden, B., Assad, A., Levy, L., Gheysens, F., (1984), The fleet size and mix vehicle routing, *Computers & Operations Research*, Vol. 11 No. 1, pp. 49-66

Golden, B., Wasil, E., Kelly, J., Ghao, I-M., (1998), The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results, in: Grainic, T., Laporte, G., (eds.), *Fleet management and logistics*, Kluwer Academic Publishers, Norwell, MA, pp. 33-56

Gomes, H. (2012), A firefly metaheuristic structural size and shape optimization with natural frequency constraints, *International journal of metaheuristics*, Vol. 2 No.1, pp. 38-55

Hassanzadeh, T., Faez, K., Seyfi, G. (2012), A speech recognition system based on structure equivalent fuzzy neural network trained by firefly algorithm, in: international conference on biomedical engineering, (ICoBE), IEEE, pp. 63-67

Imanirad, R., Yang, X.S., Yeomans, J.S. (2013), A biologically – inspired metaheuristic procedure for modeling-to-generate-alternatives, *International journal of engineering research and applications*, Vol.3 No.2, pp. 1677-1686

Imran, A., Slahi, S., Wassan, N.A. (2009), A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem, *European Journal of Operational Research*, Vol. 197 No. 2, pp. 509-518

Jacobs-Blecha, C. and Goetschalckx, M. (1992), The vehicle routing problem with backhauls: properties and solution algorithms, *Technical Report, School of Industrial and Systems Engineering*, Georgia Institute of Technology, Atlanta, Georgia. Presented at the National Transportation Research Board, January 13-15, Washington DC

Jati, G., Suyanto, (2011), Evolutionary discrete firefly algorithm for travelling salesman problem, *Adaptive and intelligent systems, Lecture notes in computer science*, pp. 393-403, Springer-Verlag, Berlin Heidelberg

Jati, G., Manurung, R., Suyanto, (2013), Discrete firefly algorithm for traveling salesman problem, *Swarm intelligence and Bio-inspired computation*, pp. 295-312

Juan, A.A., Faulin, J., Ruiz, R., Barrios, B., Caballe, S. (2010), The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem, *Applied soft computing*, Vol. 10, pp. 215-224

Kolen, A.W.J., Rinnooy Kan, A.H.G., Trienkens, H.W.J.M., (1987), Vehicle routing with time windows, *Operations Research*, Vol. 35 No. 2, pp. 266-273

Kromer, P., Abraham, A., Snasel, V., Berhan, E., Kitaw, D. (2013), On the differential evolution for vehicle routing problem, IEEE, Fifth international conference of soft computing and pattern recognition (SoCPaR 2013), Hanoi, Vietnam

- Kumbharana, S., Pandey, G. (2013), Solving travelling salesman problem using firefly algorithm, *International journal for research in science and advanced technologies*, Vol. 2 No.2, pp. 53-57
- Kwiecien, J. and Filipowicz, B. (2012), Firefly algorithm in optimization of queuing systems, *Bulletin of the Polish academy of sciences technical sciences*, Vol. 60 No. 2
- Laporte, G., (1992), The vehicle routing problem: An overview of exact and approximate algorithms, *European Journal of Operational Research*, Vol. 59 No. 3, pp. 345-358
- Laporte, G. and Louveaux, F.V. (1998), Solving stochastic routing problems with the integer L-shaped Method, in: Fleet management and logistics, Crainic, T.G. and Laporte, G. (eds.), pp. 159-167, Kluwer Academic publishers, Boston
- Laporte, G. and Osman, I.H., (1995), Routing problems: A bibliography, *Annals of Operations Research*, Vol. 61, pp. 227-262
- Laporte, G., Gendreau, M., Potvin, J., -Y., Semet, F., (2000), Classical and modern heuristics for the vehicle routing problem, *International Transactions On Operational Research*, Vol. 7, pp. 285-300
- Laporte, G. (2007), What you should know about the vehicle routing problem, *Naval Research Logistics*, Vol. 54 No.8, pp. 811-819
- Lau, H.C., Sim, M., Teo, K.M. (2003), Vehicle routing problem with time windows and a limited number of vehicles, *European Journal of Operational Research*, Vol. 148, pp. 559-569
- Leung, S.C.H., Zhang, Z., Zhang, D., Hua, X., Lim, M.K., (2013), A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints, *European Journal of Operational Research*, Vol. 225, pp. 199-210
- Li, F., Golden, B., Wasil, E. (2007), A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem, *Computations Operational Research*, Vol. 34, pp. 2734-2742
- Li, X.Y., Tian, P., Leung, S.C.H., (2009), An ant colony optimization metaheuristic hybridized with tabu search for open vehicle routing problems, *Journal of the Operational Research Society*, Vol. 60 No.7, pp. 1012-1025
- Li, X., Tian, P., Aneja, Y.P. (2010), An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *Transportational Research*, Vol. 46, pp. 1111-1127

- Lima, C.M.R.R., Goldberg, M.C., Goldberg, E.F.G., (2004), A memetic algorithm for the heterogeneous fleet vehicle routing problem, *Electronic Notes in discrete mathematics*, Vol. 18, pp. 171-176
- Lu, Q., Dessouky, M.M. (2006), A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows, *European Journal of Operational Research*, Vol. 175, No.2, pp. 672-687
- Lukasik, S., Zak, S. (2009), Firefly algorithm for continuous constrained optimization tasks, in: Nguyen, N.T., Kowalczyk, R., Chen, S.M. (eds.), *Computational collective intelligence, Semantic web, Social networks and multiagent systems*, Springer, pp. 97-106
- Luthra, J., Pal, S. (2011), A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher, in: World congress on information and communication technologies, (WICT 2011), IEEE, pp. 202-206
- Matei, O., Pop, P.C., Sas, J.L., Chira, C. (2015), An improved immigration memetic algorithm for solving the heterogeneous fixed fleet vehicle routing problem, *Neurocomputing*, Vol. 150, pp. 58-66
- Na, B., Jun, Y., Kim, B.I. (2011), Some extensions to the sweep algorithm, *International journal of advanced manufacturing technology*, Vol. 56 pp. 1057-1067
- Naji-Azimi, Z., and Salari, M. (2013), A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem, *Applied mathematical modeling*, Vol. 37, pp. 4316-4324
- Nasiri, B., Meybodi, M. (2012), Speciation based firefly algorithm for optimization in dynamic environments, *International journal of artificial intelligence*, Vol. 8 No. 12, pp. 118-132
- Osman, I.H., (1993), Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem, *Annals of operation research*, Vol. 41 No.4, pp. 421-451
- Osman, I.H., Wassan, N.A. (2002), A reactive tabu search metaheuristic for the vehicle routing problem with back-hauls, *Journal of Scheduling*, Vol. 5 No.4, pp. 263-285
- Palit, S., Sinha, S., Molla, M., Kharna, A., Kule, M. (2011), A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm, in: The second international conference on computer and communication technology (ICCT), IEEE, pp. 428-432
- Pan, F., Ye, C., Wang, K., Cao, J. (2013), Research on the vehicle routing problem with time windows using firefly algorithm, *Journal of computers*, Vol. 8 No. 9, pp. 2256-2261

- Parragh, S., Doerner, K., Hartl, R. (2008), A survey on pickup and delivery problems: Part ii: Transportation between pickup and delivery locations, *Journal für Betriebswirtschaft*, Vol. 58
- Prins, C., (2009), Two memetic algorithms for heterogeneous fleet vehicle routing problems, *Engineering Applications of Artificial Intelligence*, Vol. 22, pp. 916-928
- Prugel-Bennett, A. (2010), Benefits of a population: five mechanisms that advantage population-based algorithms, *IEEE Transactions on evolutionary computation*, Vol. 14 No. 4, pp. 500-517
- Ralphs, T., Hartman, J., Galati, M. (2001), Capacitated vehicle routing and some related problems, Rutgers university (available at: <http://neo.lcc.uma.es/vrp/wp-content/data/articles/VRP.pdf>)
- Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E. (2003), On the capacitated vehicle routing problem, *Mathematical Programming*, Vol. 94 No.2, pp. 343-359
- Renaud, J., Laporte, G., Bocton, F.F. (1996), A tabu search heuristic for the multi-depot vehicle routing problem, *Computational Operations Research*, Vol. 23 No.3, pp. 229-235
- Ribeiro, C. and Soumis, F. (1994), A column generation approach to the multiple-depot vehicle scheduling problem, *Operations Research*, Vol. 42, pp. 41-52
- Rochat, Y. and Taillard, E.D. (1995), Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics*, Vol.1 No.1, pp. 147-167
- Ropke, S. and Cordeau, J.F. (2009), Branch and cut and price for the pickup and delivery problem with time windows, *Transportation Science*, Vol. 43, pp. 267-286
- Ropke, S., Cordeau, J.F., Laporte, G. (2007), Models and branch-and-cut algorithms for pickup and delivery problems with time windows, *Networks*, Vol. 49, pp. 258-272
- Ropke, S. and Pisinger, D. (2006), An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, *Transportation Science*, Vol. 40 No.4, pp. 455-472
- Savelsbergh, M.W.P. and Sol, M. (1995), The general pickup and delivery problem, *Transportation Science*, Vol.29, pp. 17-29
- Savelsbergh, M.W.P. and Sol, M. (1998), Drive: Dynamic routing of independent vehicles, *Operational Research*, Vol. 29, pp. 474-490
- Sayadi, M., Ramezani, R., Ghaffari-Nasab, N. (2010), A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop

scheduling problems, *International journal of industrial engineering computations*, Vol. 1 No.1, pp. 1-10

Sayadi, M.K., Hafezalkotob, A., Naini, S.G.J. (2013), Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formulation, *Journal of manufacturing systems*, Vol. 32, pp. 78-84

Solomon, M. (1995), Algorithms for the vehicle routing problem with time windows, *Transportation Science*, Vol. 29 No.2, pp. 156-166

Sulaiman, M., Mustafa, M., Azmi, A., Aliman, O., Abdul Rahim, S. (2012), Optimal allocation and sizing of distributed generation in distribution system via firefly algorithm, in: IEEE international power engineering and optimization conference (PEOCO), IEEE, pp. 84-89

Taillard, E.D. (1999), A heuristic column generation method for heterogeneous fleet VRP, *RAIRO*, Vol. 33 No.1, pp. 1-14

Taillard, E.D., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y. (1997), A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, Vol. 31, pp. 170-186

Talbi, E.G. (2009), *Metaheuristics: From design to implementation*, Hoboken, New Jersey, Wiley

Tarantilis, C., Kiranoudis, C., Vassiliadis, V. (2003), A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *Journal of the Operational Research Society*, Vol. 54, pp. 65-71

Tarantilis, C., Kiranoudis, C., Vassiliadis, V. (2004), A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem, *European Journal of Operational Research*, Vol. 152 No.1, pp. 148-158

Toth, P., Vigo, D. (1999), A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls, *European Journal of Operational Research*, Vol. 113 No.3, pp. 528-543

Toth, P., Vigo, D., (2002), *The vehicle routing problem*, SIAM, Philadelphia

Xu, H., Chen, Z.L., Rajagopal, S., Arunapuram, S. (2003), Solving a practical pickup and delivery problem, *Transportation Science*, Vol. 37 No.3, pp. 347-364

Yang, X.S., (2008), *Nature-inspired metaheuristic algorithm*, Luniver Press

Yang, X.S. (2009), Firefly algorithms for multimodal optimization, in: Stochastic algorithms: Foundations and applications, Springer, pp. 169-178

- Yang, X.S., (2010a), Firefly algorithm, Stochastic test functions and design optimization, *International Journal of Bio-Inspired computation*, Vol. 2 No.2, pp. 78-84
- Yang, X.S., (2010b), *Engineering optimization: an introduction with metaheuristic applications*, Hoboken, New Jersey, Wiley
- Yang, X.S., (2010c), *Nature-inspired metaheuristic algorithms, 2nd edition*, Luniver Press
- Yang, X.S. (2010d), Firefly algorithm, levy flights and global optimization, in: Bramer, M., Ellis, R., Petridis, M. (eds.), *Research and development in intelligent systems XXVI*, Springer, pp. 209-218
- Yang, X.S., Deb, S. (2010), Eagle strategy using levy walk and firefly algorithms for stochastic optimization, in: *Nature inspired cooperative strategies for optimization, (NICSO 2010)*, pp. 101-111
- Yang, X.S. (2011a), Metaheuristic optimization: algorithm analysis and open problems, in: Pardalos, P., Rebennack, S. (eds.), *Experimental algorithms, Lecture notes in computer science*, Vol. 6630, Springer Verlag, Berlin, 2011, pp. 21-32
- Yang, X.S. (2011b), Review of meta-heuristics and generalized evolutionary walk algorithm, *International Journal of Bio-inspired computation*, Vol.3 No.2, pp. 77-84
- Yang, X.S. (2012), Efficiency analysis of swarm intelligence and randomization techniques, *Journal of computational and theoretical nanoscience*, Vol. 9 No.2, pp. 189-198
- Yang, X.S. (2013), Multiobjective firefly algorithm for continuous optimization, *Engineering with computers*, Vol. 29, pp. 175-184
- Yang, X.S. and He, X., (2013), Firefly algorithm: Recent advances and applications, *International Journal of Swarm Intelligence*, Vol.1 No. 1, pp. 36-50
- Yaohuang, G., Binglei, X., Qiang, G. (2002), Overview of stochastic vehicle routing problems, *Journal of Southwest Jiaotong university (English Edition)*, Vol. 10 No.2 , pp. 113-121
- Yu, B., Yang, Z-Z., Xie, J.X. (2011), A parallel improved ant colony optimization for multi-depot vehicle routing problem, *Journal of the Operational Research Society*, Vol. 62, pp. 183-188
- Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T. (2009), A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service, *Expert systems with applications*, Vol. 36, pp. 1070-1081

Zhang, H., Tang, J., Fung, RYK. (2011), A scatter search for multi-depot vehicle routing problem with weight-related cost, *Asian Pacific Journal of Operations Research*, Vol. 28, pp. 323-348

Παράρτημα: Απεικόνιση λύσεων των dataset

Dataset A:

A-n32-k5: 22-20-32-18-14-8-Depot-21-6-26-11-16-19-9-10-23-30-Depot-28-25-15-Depot-31-13-217-27-Depot-4-24-3-5-12-29-7

Truck order: 1-4-0-3-2

A-n33-k5: 16-13-26-27-8-9-14-33-Depot-11-28-31-6-5-21-Depot-30-17-4-10-18-23-Depot-29-7-25-3-Depot-24-19-32-2-22-15-20-12

Truck order: 3-4-1-2-0

A-n33-k6: 11-15-20-30-12-18-Depot-7-2-23-27-25-24-32-Depot-22-13-Depot-21-16-3-10-4-5-9-6-Depot-14-8-19-33-Depot-26-17-31-28-29

Truck order: 5-1-3-0-4-2

A-n34-k5: 31-25-6-5-27-Depot-15-30-28-2-16-7-8-11-Depot-14-26-18-20-12-24-Depot-9-32-29-33-22-19-Depot-3-10-4-13-23-17-34-21

Truck order: 0-3-1-2-4

A-n36-k5: 17-2-23-33-19-25-26-12-Depot-28-22-14-18-31-30-34-13-6-Depot-15-5-20-32-35-21-Depot-10-29-4-7-24-3-36-9-16-Depot-27-8-11

Truck order: 3-1-4-0-2

A-n37-k5: 22-13-10-6-5-33-4-7-Depot-1-12-2-19-20-23-14-17-Depot-36-29-32-28-31-30-15-Depot-3-24-9-11-27-8-25-35-18-26-34-Depot-21-16

Truck order: 2-0-1-4-3

A-n37-k6: 7-9-6-4-35-15-Depot-19-27-31-16-33-Depot-32-20-10-22-18-Depot-8-28-36-26-5-Depot-21-2-34-3-29-24-23-13-12-11-Depot-37-30-25-17-14

Truck order: 5-3-2-0-1-4

A-n38-k5: 22-31-30-32-38-21-Depot-19-7-5-17-26-35-20-Depot-11-13-2-4-27-24-36-34-9-Depot-25-15-3-16-14-37-18-10-Depot-33-8-23-28-12-6-29

Truck order: 1-4-0-3-2

A-n39-k5: 11-33-23-4-8-6-30-Depot-12-3-17-16-39-10-5-Depot-22-31-14-29-37-28-7-Depot-13-32-2-21-24-34-26-20-15-Depot-19-25-36-38-35-27-18-9

Truck order: 0-1-4-2-3

A-n39-k6: 11-17-5-20-3-Depot-4-25-7-2-12-Depot-6-10-29-30-27-Depot-13-39-38-32-15-36-26-34-9-8-31-Depot-37-18-24-22-19-23-35-28-33-21-Depot-14-16

Truck order: 4-5-2-3-0-1

A-n44-k6: 32-16-25-6-28-29-8-Depot-20-34-11-24-40-35-18-Depot-3-42-15-14-39-10-37-23-Depot-9-22-33-43-38-12-27-7-Depot-36-2-19-17-21-31-13-Depot-4-30-44-41-26-5

Truck order: 3-5-0-1-2-4

A-n45-k6: 38-20-31-41-18-19-27-Depot-32-39-3-26-13-40-37-16-24-Depot-6-22-34-42-9-21-14-33-Depot-25-44-35-7-2-15-Depot-36-45-12-28-30-29-Depot-8-4-11-17-5-43-23-10

Truck order: 0-5-1-3-2-4

A-n45-k7: 15-37-20-30-32-35-Depot-18-16-26-24-11-39-14-Depot-23-34-45-25-22-Depot-41-21-17-8-19-33-Depot-31-6-38-2-43-10-9-Depot-40-3-27-5-7-29-13-Depot-4-12-44-42-28-36

Truck order: 3-5-1-6-0-4-2

A-n46-k7: 22-42-30-28-2-41-35-32-21-29-Depot-39-3-38-45-8-4-6-37-Depot-24-15-31-33-23-40-Depot-12-9-34-17-25-43-13-Depot-20-16-36-26-19-11-Depot-10-46-7-44-27-14-5-Depot-18

Truck order: 3-5-1-0-2-4-6

A-n48-k7: 15-18-17-3-42-Depot-13-32-6-2-7-23-Depot-35-10-25-5-12-43-9-38-39-Depot-24-44-36-19-Depot-41-8-40-27-21-4-26-20-33-45-Depot-46-28-16-14-47-22-29-Depot-37-30-31-34-11-48

Truck order: 2-1-4-6-5-0-3

A-n53-k7: 8-29-5-23-24-44-43-3-38-Depot-31-45-37-51-50-30-11-27-41-19-39-Depot-40-6-15-22-25-42-18-26-Depot-2-10-53-14-35-12-4-Depot-34-28-47-9-Depot-48-13-17-33-16-20-49-46-Depot-36-52-32-7-21

Truck order: 2-5-4-3-6-1-0

A-n54-k7: 44-17-30-50-8-6-19-5-Depot-11-33-3-38-20-32-4-10-Depot-29-21-24-14-12-9-13-15-53-Depot-35-42-47-25-43-52-48-26-31-Depot-28-49-41-23-34-22-Depot-36-16-18-27-46-2-37-40-51-Depot-39-45-54-7

Truck order: 0-6-1-4-2-5-3

A-n55-k9: 13-11-46-51-Depot-14-24-53-25-45-2-7-Depot-42-29-20-23-31-8-5-Depot-33-40-50-36-44-Depot-9-28-55-19-49-26-Depot-43-32-27-47-15-18-38-37-30-Depot-21-3-52-12-16-Depot-22-6-54-48-10-41-17-39-Depot-34-35-4

Truck order: 1-5-6-3-8-7-2-4-0

A-n60-k9: 15-20-28-18-58-38-9-34-Depot-50-31-37-23-55-6-52-10-33-51-Depot-26-24-56-16-14-42-Depot-48-59-43-13-44-36-Depot-35-25-32-54-22-5-41-4-21-Depot-8-30-40-57-27-Depot-17-12-7-29-45-2-19-Depot-46-11-49-3-47-Depot-53-60-39

Truck order: 4-2-7-1-8-5-6-3-0

A-n61-k9: 15-7-54-9-42-57-22-Depot-34-43-33-44-27-5-23-13-Depot-40-61-12-32-53-58-51-56-Depot-31-38-46-47-55-6-Depot-29-2-17-49-59-Depot-3-11-60-50-30-37-19-36-Depot-52-24-41-28-48-35-18-45-14-Depot-4-39-16-Depot-8-25-21-26-20-10

Truck order: 4-0-8-2-5-3-1-6-7

A-n62-k8: 6-25-4-46-47-20-39-Depot-19-2-10-31-3-37-Depot-29-15-12-5-22-42-24-28-40-45-23-38-14-13-Depot-54-35-27-59-18-51-58-33-7-62-Depot-49-61-26-50-55-43-48-Depot-52-11-9-44-30-41-Depot-21-60-56-34-36-32-53-17-Depot-57-8-16

Truck order: 7-0-5-6-2-4-3-1

A-n63-k9: 24-47-2-39-32-44-62-17-Depot-16-7-38-40-43-10-4-50-Depot-41-59-58-19-5-48-53-29-Depot-49-37-28-60-30-27-Depot-12-34-54-56-15-33-23-11-Depot-45-8-26-14-61-57-Depot-13-63-52-36-51-18-Depot-21-55-35-46-6-31-20-Depot-22-42-25-9-3

Truck order: 5-3-8-1-6-0-7-2-4

A-n63-k10: 19-46-40-6-42-10-7-56-59-8-Depot-2-13-3-15-54-Depot-33-58-14-49-Depot-37-39-5-27-29-48-63-Depot-22-24-47-25-Depot-30-57-21-31-44-53-Depot-34-28-36-9-26-Depot-60-11-20-43-4-50-16-41-Depot-55-17-38-61-18-Depot-62-23-52-35-12-45-32-51

Truck order: 3-8-6-0-9-4-1-2-5-7

A-n64-k9: 10-5-50-45-22-8-51-32-56-Depot-30-16-58-23-19-Depot-42-57-63-4-Depot-33-59-55-35-38-24-52-Depot-46-28-15-36-27-40-37-41-60-54-7-Depot-20-3-6-31-12-Depot-11-14-62-25-61-9-48-47-64-17-53-Depot-39-13-44-29-2-18-26-Depot-43-21-49-34

Truck order: 6-2-3-0-8-7-5-4-1

A-n65-k9: 6-31-36-51-42-3-38-Depot-44-32-27-7-52-40-63-Depot-55-49-2-14-20-53-11-25-22-Depot-30-41-59-21-62-56-Depot-29-34-13-24-58-64-12-8-Depot-26-9-60-57-54-33-Depot-18-47-65-35-10-23-Depot-46-19-45-43-39-17-37-5-50-Depot-61-4-15-28-16-48

Truck order: 3-4-8-7-5-0-6-2-1

A-n69-k9: 24-14-46-21-42-69-Depot-27-53-63-44-Depot-55-6-17-51-18-33-38-Depot-25-40-57-49-2-54-8-Depot-35-15-43-26-48-5-61-56-28-Depot-66-22-62-65-10-19-59-20-Depot-13-67-45-16-4-52-31-50-34-3-32-Depot-29-58-12-64-47-11-37-9-41-30-Depot-68-7-36-60-39-23

Truck order: 8-6-7-0-4-1-3-5-2

A-n80-k10: 14-6-31-58-21-4-Depot-61-32-18-24-11-8-22-Depot-67-68-71-73-23-46-77-54-Depot-41-15-49-19-38-35-12-63-Depot-52-78-40-65-34-16-57-70-48-42-28-30-75-Depot-2-25-60-79-9-44-66-36-27-20-26-Depot-80-3-45-43-Depot-53-29-72-64-7-13-Depot-50-39-59-33-5-51-37-74-Depot-62-76-17-69-47-56-10-55

Truck order: 0-4-5-7-9-2-1-8-6-3

Dataset B:

B-n31-k5: 30-23-8-12-28-26-Depot-21-16-18-25-5-4-29-Depot-7-17-13-6-9-22-Depot-20-27-10-2-Depot-14-15-11-24-19-1-3

Truck order: 3-0-1-2-4

B-n34-k5: 33-32-22-30-11-Depot-15-13-28-31-17-25-6-5-Depot-14-9-Depot-12-26-3-27-29-24-16-18-19-1-Depot-8-10-23-20-4-21-7-2

Truck order: 2-4-1-0-3

B-n35-k5: 35-25-29-5-15-28-23-Depot-18-30-13-17-11-Depot-14-33-3-20-12-6-31-16-Depot-27-19-9-22-Depot-26-32-7-34-4-2-24-10-8-21

Truck order: 4-3-1-0-2

B-n38-k6: 27-17-14-38-36-3-31-Depot-15-4-29-12-23-Depot-10-24-34-19-5-6-28-Depot-22-16-21-11-2-33-Depot-32-7-20-Depot-26-8-13-30-25-35-9-18-37

Truck order: 1-3-5-0-2-4

B-n39-k5: 4-29-18-36-5-37-32-Depot-28-25-22-24-31-17-38-34-27-2-Depot-6-16-12-13-23-3-26-33-39-Depot-10-20-21-8-15-Depot-30-35-7-19-14-11-9

Truck order: 3-1-0-4-2

B-n41-k6: 2-33-13-37-6-26-Depot-15-14-11-23-32-16-Depot-30-18-29-12-35-34-7-
Depot-9-17-24-41-21-8-31-Depot-25-3-10-38-4-19-22-Depot-5-39-28-27-20-40-36

Truck order: 1-4-3-0-5-2

B-n43-k6: 42-28-26-33-30-41-Depot-6-19-20-15-29-38-24-39-40-Depot-8-13-32-9-3-
2-Depot-36-31-14-12-4-34-22-21-Depot-23-25-7-37-16-17-11-Depot-10-18-35-43-
27-5

Truck order: 4-0-5-3-1-2

B-n44-k7: 26-29-13-32-4-16-Depot-12-24-39-Depot-7-22-38-5-11-20-Depot-41-21-
36-28-44-2-18-6-Depot-14-10-31-23-43-9-42-35-Depot-37-27-25-40-15-3-34-Depot-
19-33-30-17-8

Truck order: 2-0-3-5-4-6-1

B-n45-k5: 40-44-5-34-20-42-6-Depot-4-32-39-21-36-25-19-35-8-33-3-29-Depot-16-
38-15-28-45-Depot-23-7-43-41-13-9-22-17-30-12-26-27-Depot-11-18-31-24-37-2-
10-14

Truck order: 2-1-0-3-4

B-n45-k6: 30-26-43-2-6-Depot-25-18-15-3-37-12-27-14-17-Depot-11-35-19-16-8-21-
31-44-Depot-40-9-23-29-33-24-38-20-28-5-Depot-42-22-32-4-7-Depot-36-41-34-10-
39-45-13

Truck order: 4-2-3-1-5-0

B-n50-k7: 39-10-47-27-40-13-Depot-43-4-42-34-32-23-Depot-24-36-38-12-21-9-6-
22-46-37-Depot-5-2-14-30-Depot-8-31-26-19-15-44-25-28-17-11-Depot-35-50-33-3-
49-18-20-45-16-7-48-Depot-29-41

Truck order: 3-0-6-5-1-2-4

B-n50-k8: 36-40-46-42-14-Depot-22-29-20-33-41-35-Depot-23-39-11-2-9-43-18-
Depot-6-30-Depot-26-5-19-27-48-50-31-21-3-Depot-49-8-25-17-12-44-34-38-13-45-
Depot-16-4-24-47-15-32-7-Depot-37-10-28

Truck order: 4-5-7-3-0-2-6-1

B-n51-k7: 48-23-25-31-50-3-11-18-Depot-34-49-20-4-22-10-26-47-24-Depot-9-32-
33-38-28-42-13-Depot-21-2-51-16-46-45-36-Depot-37-17-39-30-14-43-15-5-Depot-
35-12-27-8-7-Depot-6-41-19-40-29-44

Truck order: 6-3-4-5-1-2-0

B-n52-k7: 7-42-26-45-48-38-Depot-51-23-27-47-16-20-10-49-Depot-41-43-21-19-31-2-37-Depot-14-50-18-46-13-24-35-33-Depot-44-8-30-9-28-Depot-4-29-32-25-40-15-22-12-Depot-5-17-39-3-36-34-52-6-11

Truck order: 6-0-3-5-2-4-1

B-n56-k7: 23-18-9-45-47-Depot-11-29-41-6-13-10-51-39-24-52-15-44-Depot-49-36-17-53-28-8-25-Depot-43-30-16-38-5-27-22-2-Depot-48-40-42-46-54-7-19-20-26-Depot-34-14-4-Depot-3-55-31-35-32-21-56-12-50-33-37

Truck order: 3-1-6-4-2-5-0

B-n57-k7: 36-33-48-21-44-8-38-43-3-Depot-30-55-51-53-56-2-22-45-25-14-49-28-Depot-41-23-40-15-31-Depot-34-57-42-18-35-37-54-Depot-5-9-39-27-20-24-47-Depot-19-29-26-10-52-4-11-7-12-16-46-17-Depot-50-32-13-6

Truck order: 6-3-4-2-5-1-0

B-n57-k9: 48-42-37-6-54-22-5-34-Depot-35-20-53-46-44-Depot-30-38-27-45-43-14-49-39-Depot-9-16-26-52-24-21-8-36-Depot-57-23-47-51-33-Depot-50-3-17-40-15-12-Depot-31-13-56-32-Depot-7-41-4-Depot-2-55-25-28-19-29-10-18-11

Truck order: 1-0-7-6-4-5-2-8-3

B-n63-k10: 33-14-56-6-16-Depot-49-50-4-41-17-61-Depot-23-22-34-27-36-45-Depot-62-3-42-55-13-59-8-21-37-28-Depot-35-40-15-25-18-43-Depot-48-10-7-9-60-47-57-32-Depot-52-26-20-31-5-Depot-38-11-58-39-54-24-12-44-63-53-Depot-46-2-51-19-Depot-30-29

Truck order: 3-2-9-7-0-5-6-1-8-4

B-n64-k9: 48-23-37-61-49-6-5-20-Depot-43-24-50-63-47-Depot-4-36-62-29-14-18-9-42-27-Depot-51-12-22-39-Depot-8-31-17-57-30-19-13-38-Depot-34-35-41-26-40-3-64-Depot-58-21-7-54-59-45-46-Depot-11-56-52-60-15-25-16-10-33-Depot-32-28-55-44-53-2

Truck order: 8-4-3-6-7-0-2-1-5

B-n66-k9: 8-29-47-64-45-65-26-49-Depot-10-30-4-16-61-5-6-43-28-Depot-20-36-31-63-11-3-62-48-Depot-53-19-24-23-35-14-12-Depot-33-59-41-9-66-34-58-46-Depot-56-18-21-54-42-Depot-2-52-44-57-39-50-25-Depot-15-40-22-27-60-17-51-Depot-13-37-7-38-32-55

Truck order: 6-2-3-5-4-8-7-1-0

B-n67-k10: 67-39-25-30-20-12-Depot-66-51-21-50-35-7-Depot-22-26-6-49-Depot-29-64-10-45-54-5-47-15-60-Depot-14-57-16-56-55-23-53-46-13-Depot-40-43-41-65-37-2-36-18-17-Depot-63-62-31-58-4-61-Depot-44-28-52-9-38-33-3-Depot-59-32-19-27-34-48-Depot-11-8-24-42

Truck order: 3-8-0-7-1-9-6-4-2-5

B-n68-k9: 53-38-9-50-58-45-31-Depot-20-68-24-51-Depot-44-15-6-55-27-21-48-Depot-42-52-62-23-22-60-63-47-Depot-16-34-57-2-5-61-8-66-36-Depot-64-26-4-12-40-56-41-54-39-Depot-43-29-19-25-11-59-7-33-37-30-Depot-28-14-17-13-46-49-10-Depot-18-3-35-67-65-32

Truck order: 2-0-5-7-3-8-6-1-4

B-n78-k10: 59-57-28-3-39-22-78-24-12-27-Depot-67-69-37-60-10-75-63-Depot-73-72-23-47-35-13-16-Depot-48-71-30-64-70-18-Depot-9-6-31-2-25-53-68-Depot-66-58-76-50-38-77-Depot-62-40-14-34-56-36-5-29-Depot-8-33-74-4-55-21-32-46-52-Depot-41-54-65-45-7-17-19-Depot-44-51-49-20-15-42-61-11-26-43

Truck order: 6-3-7-5-1-4-2-8-0-9

Dataset P:

P-n16-k8: 3-Depot-11-13-16-Depot-7-Depot-14-10-8-Depot-15-6-Depot-12-5-Depot-4-2-Depot-9

Truck order: 4-0-5-2-6-1-3-7

P-n19-k2: 5-12-15-13-4-18-17-9-7-Depot-19-6-14-16-10-8-3-11-2

Truck order: 0-1

P-n20-k2: 5-12-16-13-4-19-18-9-14-7-Depot-20-6-15-17-10-8-3-11-2

Truck order: 0-1

P-n21-k2: 5-12-16-13-4-20-19-9-11-2-17-Depot-21-6-15-18-10-14-3-8-7

Truck order: 1-0

P-n22-k2: 17-2-11-9-19-20-4-13-16-12-5-Depot-7-8-3-14-10-22-18-15-6-21

Truck order: 0-1

P-n22-k8: 7-2-3-8-Depot-15-21-19-Depot-12-14-Depot-13-18-22-Depot-5-4-9-11-Depot-6-10-Depot-16-17-Depot-20

Truck order: 2-1-3-4-7-0-5-6

P-n23-k8: 13-9-17-Depot-4-7-Depot-3-19-18-Depot-21-20-6-Depot-8-16-Depot-2-1-Depot-10-12-15-11-Depot-5-14-22

Truck order: 3-1-2-4-7-0-5-6

P-n40-k5: 13-20-14-26-19-Depot-12-23-21-36-37-4-29-32-2-Depot-39-10-35-22-30-3-33-28-Depot-9-27-8-24-25-15-7-Depot-5-18-38-16-34-40-31-11-6-17

Truck order: 1-2-3-4-0

P-n45-k5: 19-15-26-14-42-5-Depot-12-17-3-4-29-32-27-9-28-Depot-38-20-41-43-45-6-13-Depot-31-11-40-34-16-18-8-44-25-24-7-Depot-39-10-35-22-30-21-36-37-23-2-33

Truck order: 0-2-4-3-1

P-n50-k7: 46-37-38-21-16-6-48-49-30-Depot-5-31-28-14-9-8-27-Depot-45-33-19-26-32-11-39-Depot-13-41-24-50-25-4-18-Depot-35-36-20-15-12-Depot-17-2-23-42-44-43-29-22-Depot-7-3-34-10-40-47

Truck order: 2-6-1-0-3-5-4

P-n50-k8: 5-20-9-28-14-47-Depot-41-4-33-10-40-Depot-23-22-38-21-16-30-31-Depot-13-11-12-15-36-Depot-6-48-44-43-25-50-7-Depot-17-29-49-37-46-35-Depot-3-2-34-42-24-19-26-Depot-27-8-39-32-45-18

Truck order: 2-0-3-6-7-5-1-4

P-n50-k10: 43-44-2-42-24-17-Depot-35-47-14-9-36-8-Depot-34-50-25-13-27-Depot-49-37-38-21-16-6-Depot-41-4-7-48-30-Depot-3-23-29-22-Depot-5-28-46-31-Depot-32-26-19-33-18-Depot-12-11-40-45-Depot-20-15-39-10

Truck order: 2-8-5-9-3-0-7-6-4-1

P-n51-k10: 47-10-35-31-40-11-Depot-24-27-32-30-22-51-12-Depot-7-36-37-21-33-Depot-49-2-3-4-23-Depot-14-41-20-43-45-38-18-Depot-25-44-8-9-29-Depot-6-46-34-16-5-Depot-19-15-28-Depot-26-42-48-Depot-13-50-39-17

Truck order: 0-3-8-2-5-6-4-9-7-1

P-n55-k7: 3-34-46-30-49-31-5-Depot-47-36-54-12-15-20-55-Depot-13-40-32-26-10-11-39-27-Depot-7-29-2-44-43-42-24-50-25-52-Depot-23-22-48-37-6-38-21-16-14-35-Depot-18-41-19-51-33-45-4-17-Depot-28-53-9-8

Truck order: 3-6-0-4-2-5-1

P-n55-k8: 5-15-54-39-40-10-Depot-18-51-19-26-32-11-12-Depot-27-13-41-33-45-4-52-Depot-20-8-36-9-47-35-46-Depot-53-28-16-21-48-49-23-2-44-24-Depot-3-29-43-42-25-50-17-34-Depot-7-31-30-6-22-37-38-14-55-Depot

Truck order: 0-7-1-4-2-5-6-3

P-n55-k10: 18-45-4-32-41-Depot-28-14-55-53-35-47-Depot-16-38-21-37-48-6-Depot-8-54-12-39-Depot-33-52-34-7-30-9-Depot-27-10-26-51-19-17-Depot-13-40-11-15-20-36-Depot-25-50-24-44-42-43-5-Depot-2-23-29-22-3-Depot-31-49-46

Truck order: 9-3-8-2-6-5-7-0-1-4

P-n60-k10: 18-27-20-36-9-5-Depot-3-43-44-2-34-52-Depot-12-54-15-47-Depot-10-32-40-4-45-7-Depot-33-26-19-56-51-17-Depot-14-37-21-55-28-53-35-Depot-41-22-38-16-58-46-Depot-30-6-60-48-49-31-Depot-50-25-57-24-42-23-29-Depot-13-59-11-39-8

Truck order: 1-8-3-0-4-9-6-2-7-5

P-n60-k15: 27-40-32-26-56-Depot-13-10-33-Depot-2-29-14-28-Depot-36-54-60-39-Depot-7-31-49-30-Depot-51-19-25-4-Depot-23-48-37-38-21-Depot-15-59-18-Depot-22-6-16-58-Depot-45-50-57-43-17-Depot-8-9-53-35-Depot-47-46-5-Depot-52-24-42-44-34-Depot-20-12-11-Depot-55-3-41

Truck order: 8-3-7-2-9-0-1-5-12-6-11-13-10-4-14

P-n65-k10: 63-29-31-28-53-35-Depot-58-16-21-61-37-49-5-Depot-47-39-32-51-33-Depot-13-40-45-4-19-25-57-Depot-46-15-12-8-9-Depot-6-38-48-22-62-24-34-Depot-27-36-54-60-20-55-14-30-Depot-2-23-65-43-42-44-64-50-52-Depot-18-10-26-56-11-59-Depot-3-7-17-41

Truck order: 6-0-7-5-1-2-9-4-8-3

P-n70-k10: 41-33-50-17-2-43-42-24-Depot-10-32-56-19-25-4-18-Depot-36-20-15-60-12-13-Depot-31-29-62-48-22-7-Depot-52-45-51-26-40-66-67-Depot-35-47-39-11-59-27-Depot-54-9-55-58-16-37-65-23-Depot-69-3-63-70-61-21-28-8-Depot-5-46-49-14-53-68-Depot-34-44-57-64-30-6-38

Truck order: 3-9-5-6-7-1-8-4-0-2

P-n76-k4: 53-47-35-9-46-20-36-8-12-15-60-67-66-39-54-Depot-76-69-3-74-63-29-22-62-23-65-43-44-42-57-24-50-25-45-52-18-Depot-5-68-59-73-11-32-40-10-26-56-19-51-4-17-64-2-34-7-Depot-31-75-49-48-37-70-72-61-71-21-38-6-30-16-58-14-55-28-33-41-13-27

Truck order: 3-1-0-2

P-n76-k5: 10-40-34-69-46-28-14-53-60-15-27-18-5-Depot-76-31-75-22-70-62-65-43-42-44-57-4-45-33-35-Depot-3-49-30-6-38-48-37-72-61-71-21-16-58-55-20-9-47-Depot-68-59-73-26-56-32-11-39-66-67-12-54-36-8-Depot-13-41-51-19-25-50-24-23-29-63-74-2-64-17-52-7

Truck order: 4-2-1-0-3

P-n101-k4: 27-74-73-5-40-68-3-88-60-95-100-86-54-Depot-14-7-97-96-98-93-94-62-17-87-39-15-45-92-99-38-101-43-44-16-58-42-23-24-57-76-75-22-41-59-Depot-29-13-81-55-56-26-25-30-35-79-34-10-52-21-66-67-33-91-64-12-20-65-50-37-47-48-49-83-19-53-Depot-82-80-69-78-72-36-4-77-51-2-70-32-71-31-11-63-89-8-9-84-46-18-85-6-61-90-28

Truck order: 3-0-1-2

Dataset E:

E-n22-k4: 14-12-5-4-9-11-Depot-15-22-20-17-Depot-18-21-19-16-13-Depot-10-8-6-3-2-7

Truck order: 1-0-2-3

E-n23-k3: 14-11-Depot-22-5-6-9-10-8-Depot-19-20-21-23-18-15-16-17-4-3-2-7-12-13

Truck order: 0-1-2

E-n30-k3: 26-28-27-29-25-24-6-21-Depot-22-3-4-1-5-2-20-Depot-19-15-16-13-7-17-9-14-8-12-11-10-23-18

Truck order: 0-2-1

E-n33-k4: 1-15-26-27-16-28-29-Depot-30-14-31-Depot-3-5-6-10-18-19-22-21-20-23-24-25-17-13-Depot-2-12-11-32-8-9-7-4

Truck order: 2-0-1-3

E-n51-k5: 38-45-34-46-16-18-19-26-25-Depot-47-50-10-31-11-40-35-51-6-13-Depot-39-17-22-30-21-36-37-4-3-33-2-Depot-5-43-20-41-42-14-15-44-8-24-Depot-48-12-23-29-32-27-9-49-7-28

Truck order: 1-4-2-3-0

E-n76-k7: 11-67-60-12-66-51-19-56-26-10-Depot-68-35-47-9-15-36-8-59-73-40-13-27-Depot-76-54-39-32-33-45-4-52-41-18-Depot-34-64-50-25-24-57-43-2-74-29-75-3-69-Depot-5-46-30-49-37-70-72-61-71-38-16-58-55-20-53-Depot-7-17-23-62-48-21-6-14-28-Depot-31-22-65-42-44-63

Truck order: 2-5-4-6-3-1-0

E-n76-k10: 47-35-5-75-29-9-Depot-4-25-64-43-22-48-58-53-Depot-76-69-31-3-45-40-73-11-Depot-27-13-66-10-56-51-19-50-17-Depot-68-8-54-12-39-Depot-36-20-15-67-32-59-Depot-60-55-14-16-38-21-71-37-46-Depot-44-42-65-24-57-26-33-Depot-63-23-62-70-72-61-6-49-30-28-Depot-41-7-34-74-2-52-18

Truck order: 3-0-8-2-5-9-7-1-6-4

E-n76-k14: 17-64-43-23-34-7-Depot-53-55-67-39-Depot-73-59-11-56-26-33-Depot-9-14-5-69-74-44-24-Depot-45-4-2-63-29-Depot-12-66-32-10-Depot-75-37-70-72-61-71-21-6-Depot-54-60-20-36-47-Depot-28-16-46-13-18-52-Depot-68-15-58-35-Depot-8-31-62-65-49-Depot-25-51-42-57-50-Depot-76-41-19-40-27-Depot-30-38-48-22-3

Truck order: 1-2-10-7-4-3-6-1-5-0-13-2-11-9-8