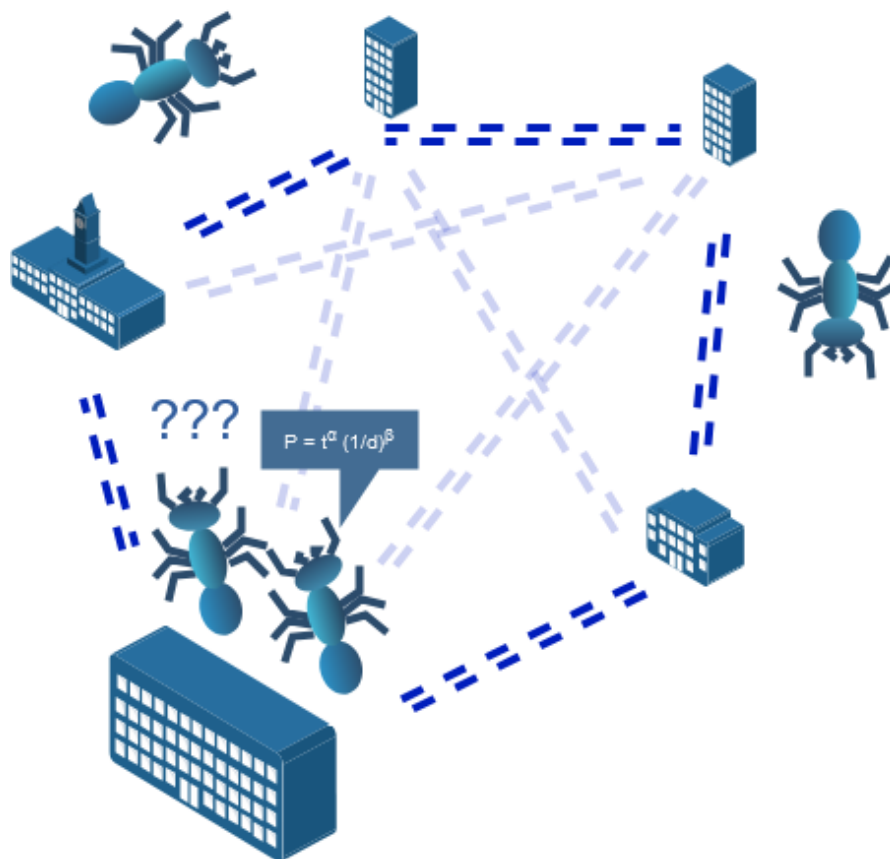


**Επίλυση του προβλήματος δρομολόγησης
συνεργατικού στόλου επίγειων και μη
επανδρωμένων εναέριων οχημάτων για παράδοση
πακέτων**



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΛΕΩΝΙΔΑΣ ΣΠΑΝΔΑΓΟΣ
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
ΜΑΡΙΝΑΚΗΣ ΙΩΑΝΝΗΣ
ΧΑΝΙΑ 2024

Ευχαριστίες

Θα ήθελα πρώτα απ' όλα να ευχαριστήσω τους γονείς μου, οι οποίοι με στήριξαν καθ' όλη τη διάρκεια των σπουδών μου και τους συμφοιτητές και φίλους μου με τους οποίους πέρασα πολλά όμορφα χρόνια στην σχολή μηχανικών παραγωγής και διοίκησης. Φυσικά θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Ιωάννη Μαρινάκη , ο οποίος μου έδειξε πόσο δημιουργικός τομέας είναι αυτός των αλγορίθμων εμπνευσμένων από την φύση, και πάντα κατάφερνε να μου εξηγήσει πολύπλοκες ιδέες με απλό και κατανοητό τρόπο. Τέλος θα ήθελα να ευχαριστήσω τον βοηθό του, κ. Νικόλαο Κυριακάκη , καθώς το πάθος του για τον προγραμματισμό με ενέπνευσε να εμβαθύνω περισσότερο στον τομέα και καθώς από την δουλειά του πήρα την ιδέα για το θέμα της διπλωματικής μου.

1. Περίληψη

Στην παρούσα εργασία εξετάζεται το πρόβλημα της δρομολόγησης συνεργατικού στόλου επίγειων και μη επανδρωμένων εναέριων οχημάτων (UAVs ή drones), το οποίο έχει διαμορφωθεί ως μια παραλλαγή του προβλήματος δρομολόγησης δύο επιπέδων με περιορισμούς χωρητικότητας και χρόνου. Αρχικά, παρουσιάζονται βασικές έννοιες, όπως η εφοδιαστική αλυσίδα, τα προβλήματα δρομολόγησης και οι μεθευρετικοί αλγόριθμοι, προκειμένου να εδραιωθεί το θεωρητικό υπόβαθρο της μελέτης. Στη συνέχεια, παρατίθενται λεπτομέρειες για το υπό εξέταση πρόβλημα, όπως οι παραδοχές του, η μαθηματική του μοντελοποίηση, καθώς και αναφορές στη σχετική βιβλιογραφία που αφορά την ενσωμάτωση μη επανδρωμένων εναέριων οχημάτων στην εφοδιαστική αλυσίδα. Παρουσιάζεται, επίσης, η στρατηγική επίλυσης του προβλήματος, συνοδευόμενη από τον κώδικα του υλοποιημένου αλγορίθμου. Ακολουθεί η ανάλυση των αποτελεσμάτων και η διαδικασία επιλογής των κατάλληλων παραμέτρων για τη βελτιστοποίηση του αλγορίθμου. Στο τέλος, διερευνάται η απόδοση μοντέλων μη επανδρωμένων εναέριων οχημάτων διαφόρων κατασκευαστών, καθώς και ενός συμβατικού συστήματος διανομής, σε ένα ρεαλιστικό σενάριο παραδόσεων μιας μικρής πόλης, με σκοπό τη συγκριτική αξιολόγηση της αποδοτικότητάς τους.

2. Εισαγωγή

Τα τελευταία χρόνια, η ζήτηση για υπηρεσίες που καλύπτουν το τελευταίο στάδιο της διανομής, γνωστό και ως «last mile delivery», έχει αυξηθεί σημαντικά, φέρνοντας στο προσκήνιο τις αδυναμίες των υφιστάμενων δικτύων μεταφοράς. Η πανδημία COVID-19 και η ραγδαία ανάπτυξη του ηλεκτρονικού εμπορίου έχουν επιδεινώσει αυτές τις πιέσεις. Αυτές οι εξελίξεις αναδεικνύουν την ανάγκη για καινοτομία και ανθεκτικότητα στην εφοδιαστική αλυσίδα. (Emrah Demir, 2022)

Παράλληλα, ο περιβαλλοντικός αντίκτυπος στον τομέα της εφοδιαστικής έχει αναδειχθεί σε σημαντικό ζήτημα, με τη ρύπανση του αέρα, τις εκπομπές αερίων του θερμοκηπίου και την κατανάλωση ενέργειας να υπογραμμίζουν τη σημασία υιοθέτησης πιο βιώσιμων λύσεων. Τόσο οι καταναλωτές όσο και οι επιχειρήσεις αναγνωρίζουν όλο και περισσότερο αυτές τις οικολογικές προκλήσεις, δίνοντας προτεραιότητα στη μείωση των αρνητικών επιπτώσεων, όπως η παραγωγή CO₂. (Anne Goodchild, 2018)

Τα drones, ως τεχνολογία για το τελευταίο στάδιο της διανομής (last mile delivery), προσφέρουν μια πολλά υποσχόμενη λύση, καθώς έχουν τη δυνατότητα να ενισχύσουν την αποτελεσματικότητα, την προσβασιμότητα και τη βιωσιμότητα των μεταφορών. Συγκριτικά με παραδοσιακές μεθόδους διανομής, είναι πιο οικονομικά, με κόστος περίπου 1,17 ευρώ ανά αποστολή για απόσταση 6,5 χιλιομέτρων, έναντι 5,10 ευρώ που απαιτεί η χρήση ηλεκτρικού φορτηγού (T. Tavares, 2019). Επιπλέον, τα drones μπορούν να φτάσουν σε περιοχές που έχουν πληγεί από φυσικές καταστροφές ή είναι απομακρυσμένες, όπου άλλα μέσα αδυνατούν να ανταποκριθούν αποτελεσματικά. (Vipul Garg a, 2023)

Αρκετές εταιρείες έχουν ήδη αρχίσει να υιοθετούν drones για εφαρμογές παράδοσης. Το πρόγραμμα "Prime Air" της Amazon στοχεύει σε παραδόσεις δεμάτων βάρους έως περίπου 2 κιλά μέσα σε 30 λεπτά, ενώ η Wing της Alphabet έχει ξεκινήσει δοκιμές για υπηρεσίες παράδοσης με drones στις Ηνωμένες Πολιτείες από το 2019. Παρομοίως, η Zipline έχει δείξει την πρακτική χρησιμότητα των drones μεταφέροντας ιατρικές προμήθειες σε περιοχές της Ρουάντα και της Γκάνας, μειώνοντας τον χρόνο μεταφοράς και σώζοντας ζωές. (Giusy Macrina, 2020)

Ωστόσο, παρόλο που τα drones παρουσιάζουν σημαντικές δυνατότητες, η περιορισμένη εμβέλειά τους και η μικρή χωρητικότητά τους αποτελούν ουσιαστικά μειονεκτήματα. Αν όμως συνδυαστούν παράλληλα με ηλεκτρικά φορτηγά, τα οποία υπερτερούν σε εμβέλεια και χωρητικότητα αλλά

υπολείπονται σε ευελιξία, προκύπτει η ευκαιρία ανάπτυξης ενός συνεργατικού στόλου που συνδυάζει τα πλεονεκτήματα και των δύο τεχνολογιών.

Μια καινοτόμα προσέγγιση θα μπορούσε να αξιοποιήσει τα ηλεκτρικά φορτηγά ως κινητές βάσεις, που μεταφέρουν drones κοντά σε ομάδες πελατών, από όπου θα πραγματοποιείται το τελικό στάδιο της παράδοσης. Αυτός ο συνδυασμός επιτρέπει την υπέρβαση των περιορισμών εμβέλειας των drones, ενώ ταυτόχρονα αξιοποιεί την ικανότητά τους να αποφεύγουν την κυκλοφοριακή συμφόρηση σε αστικά περιβάλλοντα και να παρακάμπτουν μεγάλες ορεινές διαδρομές ή φυσικά εμπόδια στις επαρχιακές περιοχές.

Επιπλέον, η τοποθέτηση καθορισμένων θέσεων στάθμευσης για τα φορτηγά θα μπορούσε να διευκολύνει τη λειτουργία του συστήματος, καθιστώντας το πρόβλημα δρομολόγησης περισσότερο διαχειρίσιμο ως προς την επίλυση του . Η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάλυση της δρομολόγησης ενός τέτοιου συνεργατικού συστήματος, εξετάζοντας τη μαθηματική μοντελοποίηση του προβλήματος, τις αλγοριθμικές μεθόδους επίλυσής του, καθώς και τις δυνατότητες υλοποίησης σε προσομοιωμένα σενάρια που βασίζονται σε δεδομένα και τεχνικές προδιαγραφές από συγκεκριμένα μη επανδρωμένα εναέρια οχήματα της αγοράς.

Πίνακας περιεχομένων

Ευχαριστίες	2
1. Περίληψη.....	3
2. Εισαγωγή	4
3. Η Εφοδιαστική Αλυσίδα	8
4. Προβλήματα δρομολόγησης	9
4.1 Το πρόβλημα του πλανοδίου πωλητή.....	10
4.2 Το Πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας(CVRP).....	10
4.3 Το Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες	11
4.4 Το Πρόβλημα δρομολόγησης οχημάτων δύο επιπέδων (2E-VRP):	12
5. Αλγόριθμοι επίλυσης προβλημάτων δρομολόγησης.....	13
5.1 Αλγόριθμοι απληστίας	13
5.2 Αλγόριθμοι τοπικής αναζήτησης	14
5.3 Μεθευρετικοί.....	15
5.3.1 Greedy Randomized Adaptive Search Procedure(GRASP)	16
5.3.2 Προσομοιωμένη ανόπτηση (Simulated Annealing ,SA)	16
5.3.3 Αλγόριθμος της Πυγολαμπίδας (Firefly algorithm, FA)	17
5.3.4 Αλγόριθμος Αποικίας Μυρμηγκιών (Ant Colony Optimization , ACO)	18
6. Μοντελοποίηση προβλήματος	22
6.1 Το πρόβλημα παράδοσης με drones	22
6.2 Παραδοχές	23
6.3 Μαθηματικό μοντέλο	24
7. Παρουσίαση κώδικα	27
7.1 Μεθοδολογία επίλυσης	27
7.2 Φιλοσοφία ανάπτυξης.....	29
7.3 Problem Maker	29
7.4 Classes	31
7.5 Grouper_3	33
7.6 Multi aco 2	35
7.7 AcoForced	37
7.8 Local Search	43

8 Αποτελέσματα	45
8.1 Απόδοση αλγορίθμου	45
8.1.1 Προβλήματα με 100 πελάτες και 5 θέσεις στάθμευσης.....	47
8.1.2 Προβλήματα με 100 πελάτες και 10 θέσεις στάθμευσης:	48
8.1.3 Προβλήματα με 150 πελάτες και 5 θέσεις στάθμευσης:	49
8.1.4 Προβλήματα με 150 πελάτες και 10 θέσεις στάθμευσης:	50
8.1.5 Προβλήματα με 200 πελάτες και 10 θέσεις στάθμευσης:	51
8.1.6 Προβλήματα με 200 πελάτες και 20 θέσεις στάθμευσης:	52
8.2 Δοκιμή διάφορων τύπων μη επανδρωμένων εναέριων οχημάτων	54
8.2.1 Χαρακτηριστικά οχημάτων	55
8.2.2 Μεθοδολογία δημιουργίας προβλημάτων	56
8.2.3 Σενάρια με 50 πελάτες	57
8.2.4 Σενάρια με 100 πελάτες	60
8.2.5 Σύγκριση με συμβατικά σενάρια	62
Συμπεράσματα	64
Βιβλιογραφία.....	65

3. Η Εφοδιαστική Αλυσίδα

Η εφοδιαστική αλυσίδα αποτελεί μια σύνθετη διαδικασία που περιλαμβάνει πολλά κινούμενα μέρη, τα οποία συνεισφέρουν άμεσα ή έμμεσα στην ικανοποίηση των απαιτήσεων του πελάτη, ενώ παράλληλα δημιουργούν αξία για τα εμπλεκόμενα μέλη. Περιλαμβάνει μεταξύ άλλων, προμηθευτές κατασκευαστές, αποθήκες, πωλητές λιανικής και φυσικά πελάτες. Είναι μια δυναμική διαδικασία και εμπεριέχει τη συνεχή ροή πληροφοριών, προϊόντων και κεφαλαίων ανάμεσα στα διάφορα στάδια της. Παρόλο που ο όρος «αλυσίδα» μπορεί να υπονοεί γραμμική δομή, στην πραγματικότητα οι περισσότερες εφοδιαστικές αλυσίδες είναι δίκτυα πολλαπλών συνδέσεων, με ροές που πραγματοποιούνται και προς τις δύο κατευθύνσεις. (Sunil Chopra, 2007)

Οι τρεις βασικοί παράγοντες που ορίζουν την απόδοση της εφοδιαστικής αλυσίδας, είναι το ολικό κέρδος, το ολικό κόστος, και ο χρονικός κύκλος. Το κέρδος προέρχεται από τους πελάτες, ενώ το κόστος από την προσπάθεια που καταβάλλεται για την μεταφορά των προϊόντων σε αυτόν. Είναι εξαιρετικά σημαντικό η αξιολόγηση της απόδοσης της αλυσίδας να γίνεται με βάση το ολικό κέρδος και το ολικό κόστος που προκύπτουν από όλη την έκταση της αλυσίδας. Αξίζει εδώ να σημειωθεί ότι υπολογίζεται ότι πολλές φορές το κόστος της μεταφοράς μπορεί να είναι το 75% του συνολικού προϋπολογισμού. Τέλος ο χρονικός κύκλος αναφέρεται στον χρόνο που χρειάζεται ένα προϊόν, ξεκινώντας σαν πρώτες ύλες να φτάσει στον πελάτη, και μάλιστα υπολογίζεται ότι μόνο το 5% αυτού αφορά την κατασκευή ή την παραγωγή του. (Ιωάννης Μαρινάκης, 2008)

Μέσα στην εφοδιαστική αλυσίδα εμπεριέχεται ένα εύρος δραστηριοτήτων, όπως η εξυπηρέτηση πελατών, η συσκευασία, η τροφοδοσία, η αντίστροφη εφοδιαστική, η πρόβλεψη ζήτησης και η χωροθέτηση εγκαταστάσεων, μεταξύ άλλων. Παρότι η λίστα δραστηριοτήτων είναι εκτενής, η παρούσα εργασία επικεντρώνεται στις μεταφορές και τη διανομή, οι οποίες αποτελούν βασικά στοιχεία της εφοδιαστικής.

Μέσω μαθηματικής μοντελοποίησης είναι δυνατό κάποιες αυτές τις δραστηριότητες ή τμήματα αυτών να αναπαρασταθούν ως μαθηματικά προβλήματα των οποίων οι λύσεις μπορούν να βελτιστοποιηθούν. Προβλήματα που μπορούν να λυθούν με αυτό τον τρόπο, μεταξύ πολλών άλλων είναι:

- **Μεταφοράς και χωροθέτησης:**

Αφορούν ζητήματα όπως ο αριθμός των αποθηκών, η βέλτιστη τοποθεσία τους, η αναγκαία χωρητικότητα και η κατανομή των φορτίων για τη μέγιστη δυνατή απόδοση ώστε να μπορέσουν να καληφθούν οι απαιτήσεις των πελατών με το μικρότερο κόστος

- **Φόρτωσης και συσκευασίας:**

Εξετάζουν τον βέλτιστο τρόπο αποθήκευσης φορτίων υπό συγκεκριμένους περιορισμούς. Ένα χαρακτηριστικό παράδειγμα αποτελεί το πρόβλημα του σακιδίου, στο οποίο ζητείται η επιλογή αντικειμένων που μεγιστοποιούν τη συνολική αξία τους, υπό τον περιορισμό ενός μέγιστου βάρους ή όγκου

- **Προγραμματισμού παραγωγής:**

Σχετίζονται με τον καθορισμό των ειδών προς παραγωγή, των ποσοτήτων τους και τον χρονικό προγραμματισμό της διαδικασίας". Στόχος είναι να καλυφθεί η ζήτηση, διατηρώντας τα αποθέματα σε βέλτιστα επίπεδα και φυσικά η μείωση του συνολικού κόστους, όπως τα κόστη αποθήκευσης και παραγωγής"

- **Μεταφορών και διανομής:**

Η επίλυση προβλημάτων στον τομέα των μεταφορών και της διανομής απαιτεί τη λήψη αποφάσεων που αφορούν την επιλογή του καταλληλότερου μέσου διανομής καθώς και τον σχεδιασμό των δρομολογίων, με στόχο τη μείωση της απόστασης και του χρόνου μεταφοράς. Οι μεταφορές διακρίνονται σε εσωτερικές, που αφορούν τη μετακίνηση αγαθών μεταξύ παραγωγικών μονάδων, και εξωτερικές, οι οποίες σχετίζονται με τη διανομή προϊόντων προς τους τελικούς καταναλωτές. Προβλήματα μεταφορών και διανομής είναι και τα προβλήματα δρομολόγησης οχημάτων. (Ιωάννης Μαρινάκης, 2008)

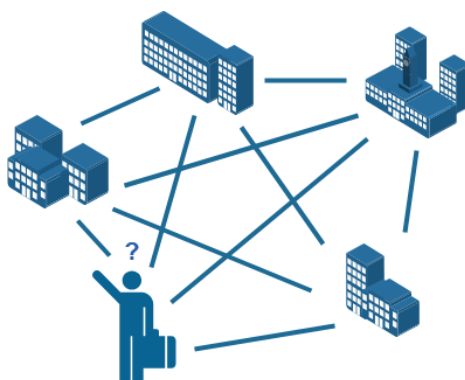
4. Προβλήματα δρομολόγησης

Τα προβλήματα δρομολόγησης αποτελούν μια υποκατηγορία των προβλημάτων μεταφορών και διανομής και είναι ένα από τα πιο σημαντικά και ευρέως μελετημένα πεδία στη συνδυαστική βελτιστοποίηση. Βασικός τους στόχος είναι η εύρεση της βέλτιστης διαδρομής για την εξυπηρέτηση ενός συνόλου πελατών. Περιλαμβάνουν μια πληθώρα διαφορετικών μοντέλων διανομής, όπως αυτά που αφορούν πολλαπλές αποθήκες, ή ταυτόχρονη παραλαβή και παράδοση, καθώς και διάφορους περιορισμούς, όπως χρονικά παράθυρα, χωρητικότητα οχημάτων, μέγιστη απόσταση ή χρόνος διαδρομής. Παράλληλα, ενσωματώνουν διαφορετικές αντικειμενικές

συναρτήσεις που σχετίζονται με διαφορετικά κριτήρια όπως η συνολική απόσταση, το κόστος, η κατανάλωση ενέργειας, ακόμα και οι εκπομπές ρύπων.

4.1 Το πρόβλημα του πλανόδιου πωλητή

Το πρόβλημα του πλανόδιου πωλητή (Travelling Salesman Problem - TSP) αποτελεί ένα από τα πλέον θεμελιώδη προβλήματα στον τομέα της συνδυαστικής. Η βασική διατύπωση αφορά έναν πωλητή, ο οποίος καλείται να επισκεφθεί έναν πεπερασμένο αριθμό κόμβων (π.χ. πελάτες) ακριβώς μία φορά, ξεκινώντας και επιστρέφοντας σε ένα προκαθορισμένο σημείο αναφοράς, όπως μια αποθήκη. Στόχος είναι να προσδιοριστεί η βέλτιστη ακολουθία επισκέψεων, η οποία ελαχιστοποιεί τη συνολική απόσταση που διανύεται.

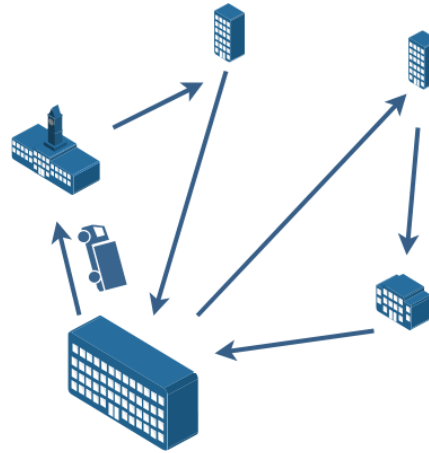


Εικόνα 4.1 Το Πρόβλημα του Πλανόδιου Πωλητή

4.2 Το Πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας(CVRP)

Μια επέκταση του προβλήματος του πλανόδιου πωλητή (TSP) είναι το Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμό Χωρητικότητας (Capacitated Vehicle Routing Problem - CVRP). Σε αυτή την παραλλαγή, προστίθεται ο περιορισμός της χωρητικότητας του οχήματος, καθώς και η ζήτηση κάθε πελάτη. Το όχημα καλείται να εξυπηρετήσει όλους τους πελάτες ακριβώς μία φορά, ικανοποιώντας τις απαιτήσεις τους. Ωστόσο, λόγω του περιορισμού της χωρητικότητας, δεν είναι πάντα εφικτό να ολοκληρωθούν όλες οι παραδόσεις σε μία μόνο διαδρομή. Κατά συνέπεια, απαιτούνται επιστροφές στην αποθήκη για ανεφοδιασμό. Οι πολλαπλές διαδρομές που

σχηματίζονται μπορούν να κατανεμηθούν σε ένα ή περισσότερα οχήματα. Στόχος παραμένει η ελαχιστοποίηση είτε της συνολικής απόστασης που διανύεται είτε του συνολικού χρόνου που απαιτείται.

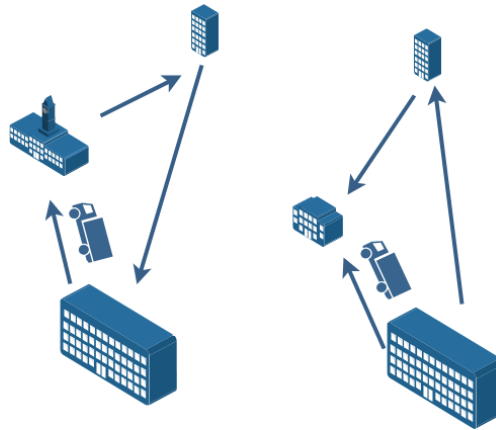


Εικόνα 4.2 CVRP

4.3 Το Πρόβλημα δρομολόγησης οχημάτων με πολλαπλές αποθήκες

Στο Πρόβλημα Δρομολόγησης Οχημάτων με Πολλαπλές Αποθήκες (Multi-Depot Vehicle Routing Problem - MDVRP), υπάρχουν περισσότερες από μία κεντρικές αποθήκες, από τις οποίες μπορούν να ξεκινήσουν και να επιστρέψουν τα οχήματα. Η βασική πρόκληση εδώ έγκειται στον προσδιορισμό του τρόπου με τον οποίο θα κατανεμηθούν οι πελάτες στις διαθέσιμες αποθήκες, καθώς και στον καθορισμό της αποθήκης στην οποία θα επιστρέψουν τα οχήματα μετά την ολοκλήρωση των διαδρομών τους.

Η συνήθης διαδικασία επίλυσης του προβλήματος περιλαμβάνει δύο φάσεις: Στην πρώτη φάση αποφασίζεται από ποια αποθήκη θα εξυπηρετηθεί κάθε πελάτης, λαμβάνοντας υπόψη παράγοντες όπως η απόσταση και η ζήτηση. Στη δεύτερη φάση, το πρόβλημα διασπάται σε μικρότερα, ανεξάρτητα προβλήματα δρομολόγησης, ένα για κάθε αποθήκη. Κάθε πρόβλημα επιλύεται με στόχο την ελαχιστοποίηση της συνολικής απόστασης ή του χρόνου.



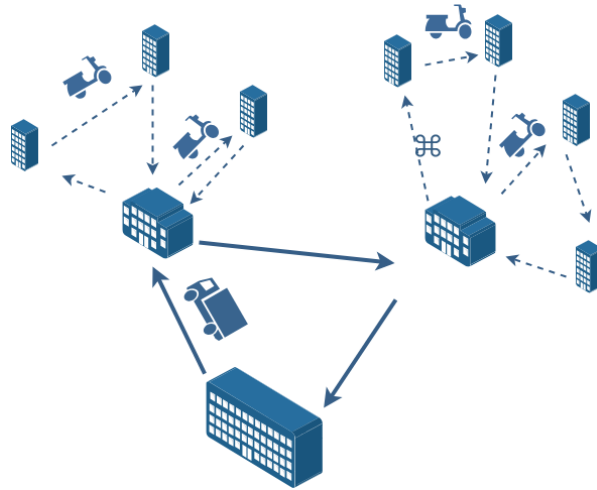
Εικόνα 4.3 MDVRP

4.4 Το Πρόβλημα δρομολόγησης οχημάτων δύο επιπέδων (2E-VRP):

Η χρήση αποκλειστικά ενός τύπου οχήματος για τη διανομή απευθείας από την κεντρική αποθήκη συχνά δεν είναι επαρκώς αποδοτική. Η ενσωμάτωση ενδιάμεσων αποθηκών (satellites) και η αξιοποίηση διαφορετικών τύπων οχημάτων (μεγαλύτερων και μικρότερων) δύναται να προσφέρει σημαντικά λειτουργικά πλεονεκτήματα.

Στο συγκεκριμένο πρόβλημα, η διαδικασία διανομής πραγματοποιείται σε δύο στάδια: Αρχικά το φορτίο μεταφέρεται από την κεντρική αποθήκη στις ενδιάμεσες αποθήκες με μεγαλύτερα οχήματα με στόχο την στρατηγική τοποθέτηση του αποθέματος πιο κοντά στους πελάτες. Στην συνέχεια η τελική παράδοση στους πελάτες πραγματοποιείται με τη χρήση μικρότερων οχημάτων από τις ενδιάμεσες αποθήκες. (U. Breunig, 2016)

Το πρόβλημα αυτό παρουσιάζει πολλές παραλλαγές, που ενσωματώνουν διαφορετικούς περιορισμούς, όπως η χωρητικότητα των οχημάτων, ο μέγιστος επιτρεπτός χρόνος διαδρομής και άλλοι παράγοντες. Σε γενικές γραμμές, δεν επιτρέπονται οι απευθείας παραδόσεις από την κεντρική αποθήκη στους πελάτες, ούτε οι πολλαπλές επισκέψεις στους πελάτες από διαφορετικά οχήματα. Αντιθέτως, επιτρέπονται οι πολλαπλές επισκέψεις στις ενδιάμεσες αποθήκες.



ΦΕικώνα 4.4 2E-VRP

5. Αλγόριθμοι επίλυσης προβλημάτων δρομολόγησης

Τα προβλήματα δρομολόγησης οχημάτων χαρακτηρίζονται ως NP-hard, γεγονός που τα καθιστά ιδιαίτερα απαιτητικά από υπολογιστικής άποψης. Ένα πρόβλημα ανήκει στην κατηγορία NP-hard όταν η επίλυσή του απαιτεί χρόνο που αυξάνεται εκθετικά με το μέγεθος των δεδομένων. Αυτό σημαίνει ότι η πλήρης εξερεύνηση όλων των πιθανών λύσεων καθίσταται πρακτικά αδύνατη ακόμα και για σχετικά μικρά προβλήματα.

Για αυτόν τον λόγο, συχνά εφαρμόζονται ευρετικοί αλγόριθμοι, οι οποίοι παρέχουν ικανοποιητικά αποτελέσματα σε σύντομο χρονικό διάστημα, αξιοποιώντας απλουστευμένες μεθόδους ή και τυχαίες διαδικασίες αναζήτησης. Πρόκειται για μια ευρεία κατηγορία αλγορίθμων, που περιλαμβάνει, μεταξύ άλλων, τους αλγορίθμους απληστίας και τους αλγορίθμους τοπικής αναζήτησης.

5.1 Αλγόριθμοι απληστίας

Αυτοί οι αλγόριθμοι θεωρούνται «Μυωπικοί» καθώς κοιτούν μόνο μπροστά χωρίς να κρατάνε πληροφορία από προηγούμενες επαναλήψεις ή να εξερευνούν τον χώρο των λύσεων. Η έννοια της απληστίας έγκειται στο γεγονός ότι κατασκευάζουν μια λύση (ή βελτιώνουν μια) παίρνοντας αποφάσεις με κάποιο κριτήριο, το οποίο τουλάχιστον για το πλαίσιο της δεδομένης στιγμής οδηγεί σε μια σχετικά καλή απόφαση. (Ιωάννης Μαρινάκης, 2008)

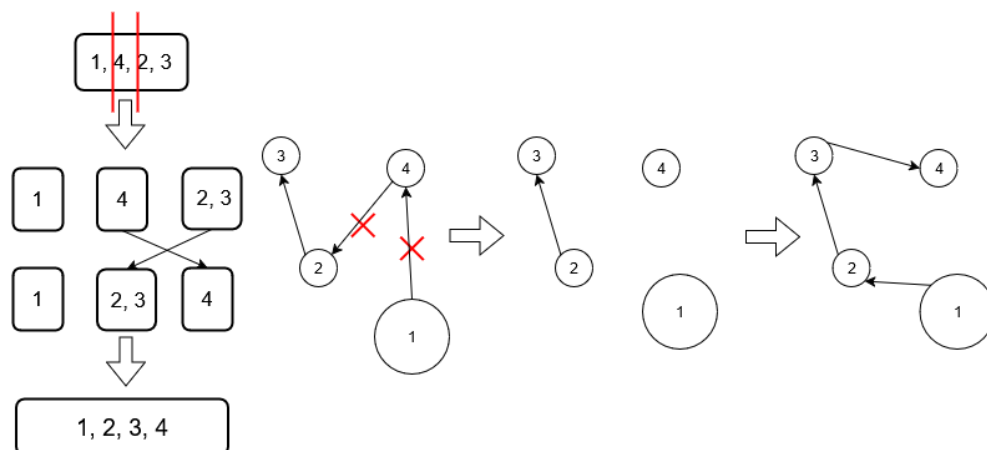
Ο πιο κλασσικός αλγόριθμος απληστίας είναι ο πλησιέστερος γείτονας. Η αρχή λειτουργίας του είναι πολύ απλή. Στην περίπτωση του προβλήματος

του πλανόδιου πωλητή, η λύση αναπτύσσεται κατασκευαστικά, επιλέγοντας πάντα τον πιο κοντινό πελάτη τη δεδομένη στιγμή.

5.2 Αλγόριθμοι τοπικής αναζήτησης

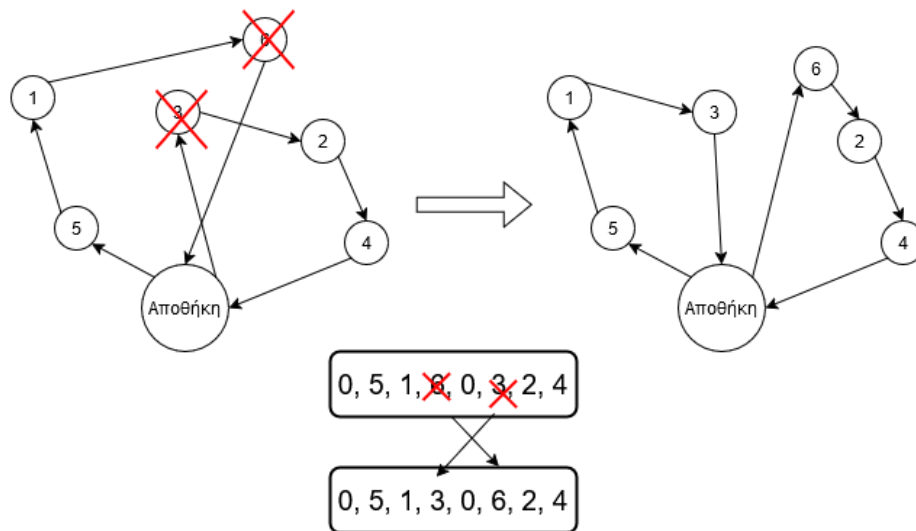
Οι αλγόριθμοι τοπικής αναζήτησης έχουν ως στόχο να βελτιώσουν μια ήδη υπάρχουσα εφικτή λύση, δοκιμάζοντας διάφορες αλλαγές και μετασχηματισμούς (αναζητώντας την γειτονιά). Είναι πολύ απλοί αλλά εκπληκτικά αποτελεσματικοί. Κάποιοι πολύ βασικοί αλγόριθμοι είναι οι εξής:

- 2-Opt:
Ουσιαστικά ο αλγόριθμος 2-Opt περιλαμβάνει την διαγραφή δύο ακμών και την επανασύνδεση τους. Δοκιμάζει όλες τις δυνατές επανασυνδέσεις και κρατάει την καλύτερη. Είναι πιο εύκολα κατανοητό αν αναπαρασταθεί σαν το «κόψιμο» της διαδρομής σε δυο κομμάτια και την επανασύνδεσή της.



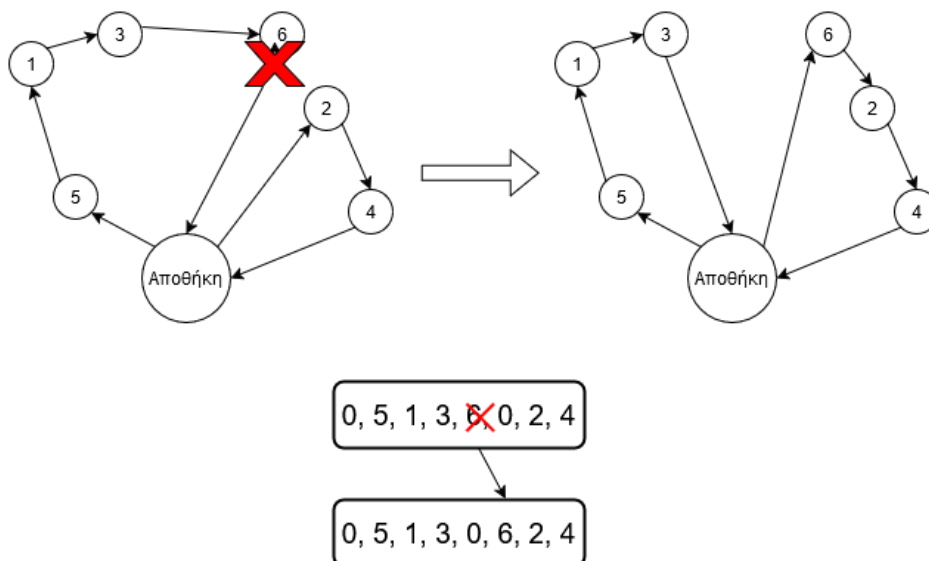
Εικόνα 5.1 2-Opt, οπτικοποίηση αριστερά σε σειρά επίσκεψης , δεξιά σε μορφή γράφων

- 3-Opt:
Είναι παρόμοιος με τον 2-Opt αλλά αυτήν την φορά διαγράφονται και αναδημιουργούνται 3 ακμές
- 1-1 exchange:
Μια κίνηση 1-1 exchange περιγράφει την ανταλλαγή ενός πελάτη σε μια διαδρομή με ενός πελάτη σε μια άλλη.



Εικόνα 5.2 1-1 Exchange

- 1-0 relocate:
Μια κίνηση 1-0 relocate είναι να επανατοποθετηθεί ένας πελάτης από μια διαδρομή σε μια άλλη.



Εικόνα 5.3 1-0 Relocate

5.3 Μεθευρετικοί

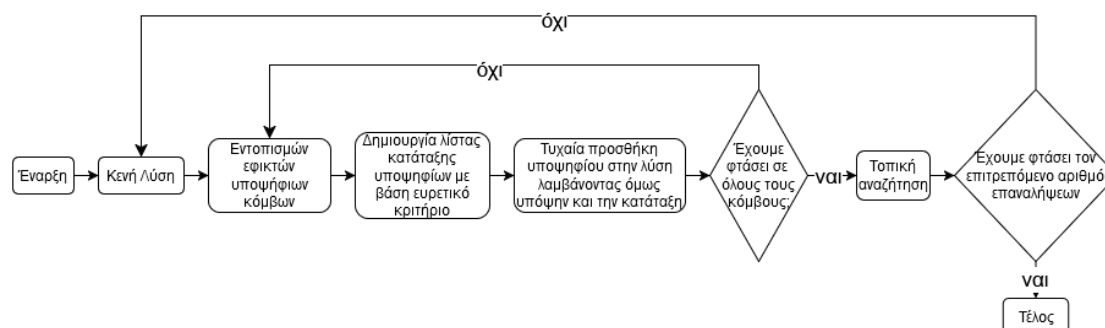
Οι Μεθευρετικοί αλγόριθμοι επίσης επιδιώκουν την εύρεση ικανοποιητικών, αν και όχι απαραίτητα βέλτιστων, λύσεων σε σύνθετα προβλήματα εντός λογικού χρόνου. Σε αντίθεση με τους απλούς ευρετικούς αλγόριθμους όμως, εφαρμόζουν στρατηγικές υψηλότερου επιπέδου, λειτουργώντας

επαναληπτικά και ενσωματώνοντας μηχανισμούς που τους επιτρέπουν να αποδρούν από τοπικά βέλτιστα. Επιπλέον, χρησιμοποιούν μηχανισμούς μνήμης για την αποθήκευση πληροφοριών από λύσεις που παρουσιάζουν υψηλή απόδοση, καθώς και τεχνικές που διευκολύνουν την παράλληλη αναζήτηση σε πολλαπλά τοπικά βέλτιστα, βελτιώνοντας έτσι την προσέγγισή τους προς το ολικό. (Michel Gendreau, 2003)

Πολλές φορές είναι βασισμένοι στην τυχαιότητα, όπως η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (GRASP), εμπνευσμένοι από φυσικά φαινόμενα όπως ο Αλγόριθμος προσομοιωμένης ανόπτησης (Simulated - Annealing), ή εμπνευσμένοι από την φύση όπως η αποικία μυρμηγκιών και ο αλγόριθμος της πυγολαμπίδας. Παρακάτω θα παρουσιαστούν μερικοί από αυτούς τους αλγορίθμους:

5.3.1 Greedy Randomized Adaptive Search Procedure(GRASP)

Ο GRASP δουλεύει σε δύο στάδια. Αρχικά κατασκευάζει μια εφικτή λύση προσθέτοντας κόμβους με μια τυχαιοποιημένη συνάρτηση απληστίας. Ύστερα η λύση αυτή βελτιώνεται με τοπική αναζήτηση. Η διαδικασία αυτή επαναλαμβάνεται πολλές φορές και αποθηκεύεται η καλύτερη λύση που έχει βρεθεί μέχρι τότε. Η λειτουργία του φαίνεται πιο αναλυτικά στο παρακάτω διάγραμμα. (Ιωάννης Μαρινάκης, 2008)



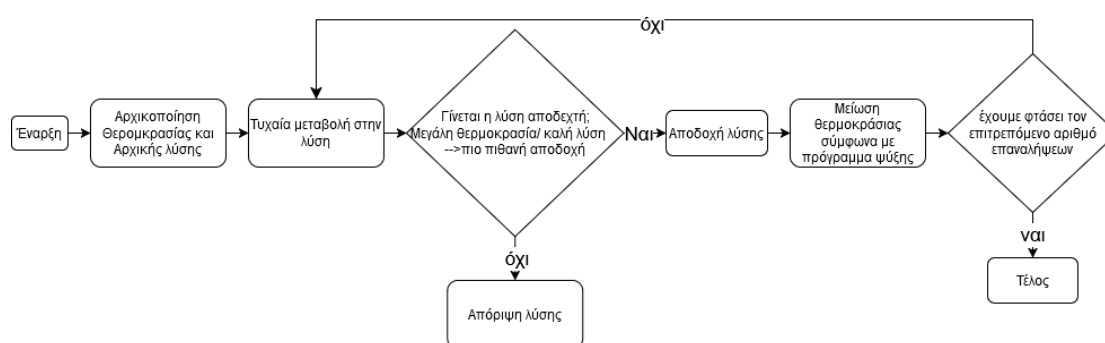
Εικόνα 5.4 Διάγραμμα ροής GRASP

5.3.2 Προσομοιωμένη ανόπτηση (Simulated Annealing ,SA)

Ο κεντρικός μηχανισμός της SA είναι ότι καθώς ο χώρος των λύσεων εξερευνάται, επιτρέπονται, υπό κάποιες συνθήκες, μεταβάσεις σε χειρότερες λύσεις (hill climb). Αυτό δίνει την δυνατότητα στον αλγόριθμο να αποδρά από τοπικά ελάχιστα. Το όνομα του αλγορίθμου σχετίζεται λοιπόν με αυτόν τον μηχανισμό αποδοχής «κακών» λύσεων και το πώς είναι εμπνευσμένος από την διαδικασία της ανόπτησης. (Michel Gendreau, 2003)

Η ανόπτηση είναι μια μέθοδος μέσω της οποίας ένα υλικό βελτιώνεται ως προς την δομή του μέσω της θέρμανσης και της αργής ψύξης του. Η υψηλή θερμοκρασία δίνει την δυνατότητα στα μόρια να φύγουν από την δομή τους και καθώς η θερμοκρασία μειώνεται να σταθεροποιηθούν σε θέσεις χαμηλότερες ενέργειας. Έτσι δημιουργούνται πιο ενεργειακά σταθερές και άρα πιο ανθεκτικές μοριακές δομές.

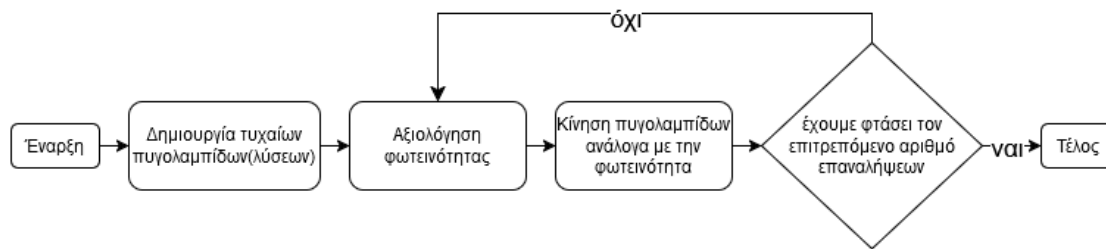
Αντίστοιχα ο SA ξεκινά από μια αρχική λύση, την οποία μεταβάλλει σε κάθε επανάληψη με τελεστές τοπικής αναζήτησης, και φυσικά κρατάει την καλύτερη λύση που έχει εντοπίσει ως τώρα. Παράλληλα όμως επειδή στην αρχή βρίσκεται σε μια κατάσταση υψηλής θερμοκρασίας είναι πιθανό να «δεχτεί» να κρατήσει και μια κακή λύση (χειρότερη από την ως τότε καλύτερη). Σε επόμενες επαναλήψεις που η θερμοκρασία όλο και μειώνεται η αποδοχή γίνεται όλο και πιο δύσκολη.



Εικόνα 5.5 Διάγραμμα ροής SA

5.3.3 Αλγόριθμος της Πυγολαμπίδας (Firefly algorithm, FA)

Ο αλγόριθμος της πυγολαμπίδας βασίζεται στην ιδέα ότι οι πυγολαμπίδες επικοινωνούν η μια με την άλλη με το φως τους. Στο πλαίσιο της βελτιστοποίησης, η θέση της πυγολαμπίδας αντιπροσωπεύει μια λύση και η φωτεινότητα την τιμή της αντικειμενικής συνάρτησης. Σε κάθε επανάληψη οι πυγολαμπίδες (λύσεις) κινούνται προς τις πιο φωτεινές (με χαμηλότερο κόστος). Η έλξη μεταξύ τους είναι συνάρτηση της φωτεινότητας και της απόστασής τους. (Yang, 2009)



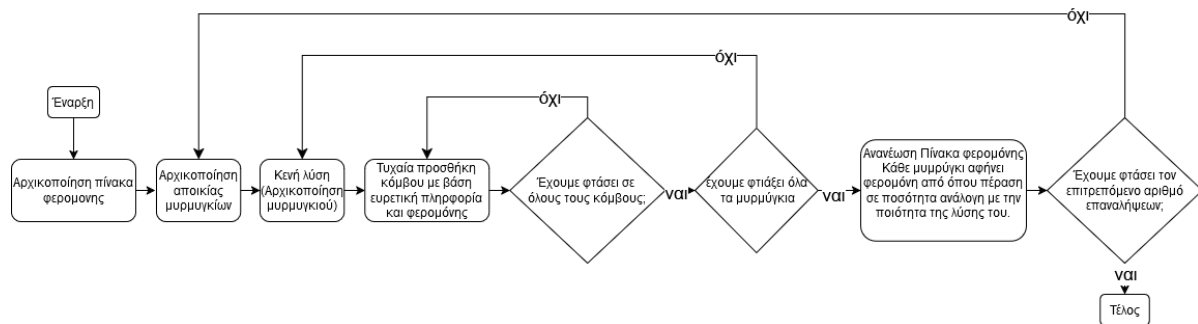
Εικόνα 5.6 Διάγραμμα ροής FA

Ο συγκεκριμένος αλγόριθμος δουλεύει καλύτερα σε προβλήματα συνεχούς βελτιστοποίησης καθώς κάθε μεταβλητή μπορεί να αντιστοιχεί σε μια διάσταση στον n -διάστατο χώρο στο οποίο βρίσκονται οι πυγολαμπίδες. Ακόμα σε ένα τέτοιο χώρο είναι πολύ εύκολο να προσδιοριστεί η έννοια της κίνησης και της απόστασης.

Στο χώρο των λύσεων των προβλημάτων συνδυαστικής βελτιστοποίησης αυτές οι αναπαραστάσεις είναι πιο δύσκολες. Απάντηση σε αυτό το πρόβλημα είναι συναρτήσεις που μετατρέπουν τις λύσεις από θέση στον χώρο, σε διαδρομή και αντίστροφα ή διαφορετικές αναπαραστάσεις της κίνησης, της θέσης και της απόστασης (πχ Hamming distance). Αν αυτά τα προβλήματα λυθούν ο αλγόριθμος αυτός μπορεί να αποδώσει εξαιρετικά αποτελέσματα καθώς έχει το πλεονέκτημα να εξερευνά πολλά «σημεία» στον χώρο των λύσεων ταυτόχρονα.

5.3.4 Αλγόριθμος Αποικίας Μυρμηγκιών (Ant Colony Optimization , ACO)

Πολλά είδη μυρμηγκιών παρουσιάζουν συμπεριφορές αναζήτησης τροφής, οι οποίες βασίζονται στην έμμεση επικοινωνία μέσω απόθεσης φερομόνων . Πιο συγκεκριμένα η φερομόνη που εναποτίθεται σχηματίζει διαδρομές, τις οποίες τα μυρμήγκια εντοπίζουν με την όσφρησή τους και τείνουν να ακολουθούν. Στο πλαίσιο του ACO, ένα τεχνητό μυρμήγκι κατασκευάζει μια λύση στοχαστικά με βάση όμως την ποσότητα φερομόνης αλλά και ευρετική πληροφορία (πχ οι κόμβοι με πιο μικρό κόστος μετάβασης από τον τρέχοντα κόμβο είναι πιο πιθανό να επιλεγούν). Η πληροφορία της φερομόνης βρίσκεται σε ένα γράφο και αντιπροσωπεύει τις πιθανότητες μετάβασης των κόμβων μεταξύ τους. (Marco Dorigo, 2004).



Εικόνα 5.7 Διάγραμμα ροής ACO

Ο τύπος που χρησιμοποιείται για τον υπολογισμό των πιθανοτήτων μεταβάσεων μεταξύ των κόμβων είναι ο:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{k \in \text{επιφικτές}} \tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}$$

Στον τύπο αυτό το P_{ij} αντιπροσωπεύει τις πιθανότητες μετάβασης από τον κόμβο i στον j . Το τ_{ij} συμβολίζει το επίπεδο φερομόνης στην ακμή (i, j) , που αποτελεί μέτρο της ελκυστικότητας της. Η ευρετική πληροφορία η_{ij} , σχετίζεται με την απόσταση d_{ij} ανάμεσα στους κόμβους i και j και υπολογίζεται από τον τύπο $\eta_{ij} = \frac{1}{d_{ij}}$. Οι παράμετροι α και β ρυθμίζουν τη σημασία της φερομόνης και της ευρετικής πληροφορίας αντίστοιχα, κατά τη διαδικασία επιλογής διαδρομών από τα μυρμήγκια.

Η σωστή επιλογή των παραμέτρων α και β , επηρεάζει άμεσα την απόδοση του αλγορίθμου. Όταν αυξάνεται το α δίνεται μεγαλύτερη έμφαση στη φερομόνη. Αυτό μπορεί να είναι ωφέλιμο για την εκμετάλλευση καλών λύσεων, αλλά η υπερβολική αύξηση του α μπορεί να οδηγήσει σε πρόωρη σύγκλιση σε τοπικά ελάχιστα. Αντίθετα, η αύξηση του β ενισχύει τη σημασία της ευρετικής πληροφορίας (οι μικρότερες αποστάσεις), μεταβάλλοντας τον αλγόριθμο στο να προσομοιάζει περισσότερο την λειτουργία ενός αλγορίθμου απληστίας. Φυσικά, όταν το β είναι υπερβολικά μεγάλο, η επίδραση της φερομόνης μειώνεται.

Υπάρχουν πολλές παραλλαγές του αλγορίθμου που αξίζει να παρουσιαστούν:

- Σύστημα Μυρμηγκιών (Ant System - AS). Είναι ο πρώτος αλγόριθμος αποικίας μυρμηγκιών. Μια δημοφιλή τροποποίηση του, είναι η στρατηγική ελίτ (EAS), όπου τα μυρμήγκια που βρίσκουν καλύτερες

λύσεις έχουν τη δυνατότητα να εναποθέσουν περισσότερη φερομόνη. Αυτή η τακτική έχει φανεί να αυξάνει την απόδοση του.

- Σύστημα Αποικίας Μυρμηγκιών (Ant Colony System - ACS) . Είναι η ενισχυμένη εκδοχή του AS (Dorigo, Gambardella). Η κύρια διαφορά είναι ότι ο ACS εισάγει δύο κανόνες ενημέρωσης φερομόνης: τοπική (στην κατασκευή λύσεων) και ολική ενημέρωση (στο τέλος κάθε επανάληψης). Επίσης η στοχαστική απόφαση βασίζεται σε μια τυχαία μεταβλητή, η οποία καθορίζει αν θα επιλεγεί το καλύτερο δυνατό στοιχείο.
- Σύστημα Μυρμηγκιών MAX-MIN (MMAS) Το MMAS, (Stützle και Hoos), Στην οποία μόνο το καλύτερο μυρμήγκι εναποθέτει φερομόνη και οι μέγιστες και ελάχιστες τιμές φερομόνης περιορίζονται. Όλες οι τιμές του πίνακα φερομόνης αρχικοποιούνται στο μέγιστο και αρχικοποιούνται ξανά στο μέγιστο όταν παρατηρείται στασιμότητα στην βελτίωση των λύσεων.
- Σύστημα Μυρμηγκιών με Βαθμολογία (ASrank). Ο ASrank ταξινομεί όλες τις λύσεις με βάση την ποιότητα τους. Η ποσότητα φερομόνης που εναποτίθεται εξαρτάται από την ποιότητα της λύσης, με τις καλύτερες λύσεις να εναποθέτουν περισσότερη φερομόνη.
- Αναδρομική Βελτιστοποίηση Αποικίας Μυρμηγκιών (Recursive Ant Colony Optimization). Αυτή η εκδοχή διαιρεί τον συνολικό χώρο αναζήτησης σε υπό-περιοχές, επιλύει το πρόβλημα σε αυτές, συγκρίνει τα αποτελέσματα και προωθεί τα καλύτερα για το επόμενο επίπεδο.
- AntNet. Το AntNet (Di Caro, Dorigo) εισάγει τα "προωθητικά" και "επιστρεφόμενα" μυρμήγκια. Τα προωθητικά μυρμήγκια επιλέγουν την επόμενη διαδρομή τυχαία αλλά λαμβάνοντας υπόψιν και ευρετική πληροφορία , ενώ τα επιστρεφόμενα μυρμήγκια ακολουθούν την αντίστροφη διαδρομή προς την πηγή, αφήνοντας φερομόνες σε κάθε σύνδεσμο. (O. Deepa, 2016)

Στην συγκεκριμένη διπλωματική υιοθετείται μια εκδοχή του αλγορίθμου, η οποία έχει πολλά κοινά γνωρίσματα με τον ASrank. Πιο συγκεκριμένα, μόνο τα καλύτερα μυρμήγκια εναποθέτουν φερομόνη σε ποσότητα που ισοδυναμεί με το μέσο κόστος όλων των μυρμηγκιών διαιρεμένο με το κόστος του τρέχοντος. Κατά συνέπεια, όσο μικρότερο κόστος έχουν τόσο περισσότερη φερομόνη τοποθετούν. Αντίθετα ο ASrank ενσωματώνει μια ακόμα λειτουργία σύμφωνα με την οποία ανανεώνει την φερομόνη και με το

καλύτερο μυρμήγκι που έχει βρεθεί σε όλες τις επαναλήψεις. Η λειτουργία του θα αναλυθεί περισσότερο στο διάγραμμα ροής στο κεφάλαιο παρουσίαση κώδικα.

6. Μοντελοποίηση προβλήματος

Σε αυτό το κεφάλαιο θα αναφερθούν ιδέες από τις μελέτες διάφορων ερευνητών, οι παραδοχές που πάρθηκαν για το μοντέλο της παρούσας εργασίας καθώς και η σχετική μαθηματική μοντελοποίηση σαν πρόβλημα ελαχιστοποίησης απόστασης.

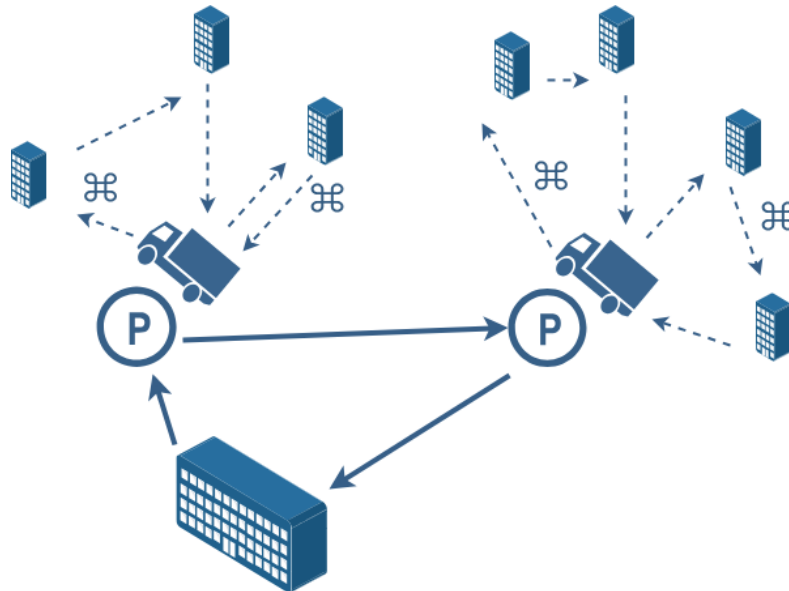
6.1 Το πρόβλημα παράδοσης με drones

Όπως αναφέρθηκε στην εισαγωγή, η ενσωμάτωση των μη επανδρωμένων οχημάτων στην εφοδιαστική αλυσίδα ίσως είναι μια λύση στο ποσοστό κόστος του last mile delivery. Διάφορα μοντέλα έχουν παρουσιαστεί τα οποία χρησιμοποιούν διαφορετικές παραδοχές, περιορισμούς και αντικειμενικές συναρτήσεις. Η μελέτη των (Zhen et al, 2019) για παράδειγμα, λαμβάνει υπόψη για την κατανάλωση ενέργειας των drones, την απόσταση και το φορτίο αλλά και το ύψος πτήσης. Επιπλέον οι Raj και Murray επικεντρώνονται στη μεταβαλλόμενη ταχύτητα των drones και το πώς αυτή επηρεάζει την αποδοτικότητα των παραδόσεων σε πυκνοκατοικημένες περιοχές (Raj, 2020). Επιπρόσθετα ο (Shahzaad, 2021) και η ομάδα του θέτουν περιορισμούς για ζώνες απαγόρευσης πτήσεων, συνθήκες ανέμου και αξιοποιούν ταράτσες κτιρίων ως σημεία φόρτισης ή παραδόσεων.

Ακόμα έχουν προταθεί πολλά διαφορετικά μοντέλα συνεργατικών στόλων επίγειων και εναέριων οχημάτων. Ιδιαίτερο ενδιαφέρον έχει μια δημοσίευση στην οποία το drone χρησιμοποιείται για την ανατροφοδότηση του φορτηγού κατά τη διάρκεια της διαδρομής του, καθώς ίσως είναι το πιο ρεαλιστικό σενάριο με βάση την τωρινή τεχνολογία (Pina-Pardo, 2021). Ακόμα ένα πολλά υποσχόμενο μοντέλο είναι αυτό στο οποίο ένα φορτηγό λειτουργεί σαν κινητή βάση για τα drones τα οποία επισκέπτονται τους πελάτες, όπως η μελέτη του Kitjacharoenchai και των συνεργατών του που προτείνουν ένα ευέλικτο σύστημα παράδοσης, όπου τα drones μπορούν να επιστρέφουν σε οποιοδήποτε φορτηγό (Kitjacharoenchai, 2020). Ένα ακόμα τέτοιο παράδειγμα είναι η δημοσίευση των (Nikolaos A. Kyriakakis, 2022), όπου τα φορτηγά σταματάνε σε προκαθορισμένες θέσεις στάθμευσης, απελευθερώνουν τα UAVs και αναμένουν την επιστροφή τους. Μάλιστα στην συγκεκριμένη εκδοχή η ενεργειακή κατανάλωση η οποία ελαχιστοποιείται, υπολογίζεται με ένα πολύ κομψό τρόπο, ως η μετακίνηση του οχήματος επί το βάρος του, λαμβάνοντας έτσι υπόψη και το φορτίο.

Η συγκεκριμένη εργασία αποτελεί μια απλοποιημένη εκδοχή της τελευταίας. Οι διαφοροποιήσεις εντοπίζονται στα εξής σημεία: ο στόχος είναι η ελαχιστοποίηση της απόστασης, η ενέργεια υπολογίζεται ως συνάρτηση της

κατανάλωσης του οχήματος και της απόστασης, και δεν υφίσταται περιορισμός στον αριθμό των πακέτων που μπορεί να μεταφέρει ένα εναέριο όχημα, παρά μόνο στο βάρος και στην εμβέλεια. Οι παραδοχές θα αναλυθούν εκτενέστερα στη συνέχεια.



Εικόνα 6.1 Drone-Truck CVRP

6.2 Παραδοχές

Το πρόβλημα προσεγγίστηκε με βάση τις παρακάτω υποθέσεις:

- Τα drones διαθέτουν μηχανισμό άμεσης αντικατάστασης μπαταριών με πλήρως φορτισμένες μονάδες κατά την επιστροφή τους στο όχημα μεταφοράς.
- Τα φορτηγά διαθέτει σύστημα ακαριαίας φόρτισης κατά την άφιξη στην αποθήκη.
- Τα φορτηγά αδυνατούν να επισκεφτούν κατευθείαν τους πελάτες
- Τα drones δεν δύνανται να εκκινήσουν απευθείας από την αποθήκη.
- Τα drones επιστρέφουν αποκλειστικά στο όχημα από το οποίο αρχικώς απελευθερώθηκαν.
- Κάθε θέση στάθμευσης δέχεται ακριβώς μια επίσκεψη
- Το φορτίο των drone δεν περιορίζεται από τον αριθμό ή των όγκο των πακέτων αλλά μόνο από το βάρος.
- Δεν λαμβάνεται υπ' όψιν ο χρόνος απογείωσης, προσγείωσης και παράδοσης πακέτου

6.3 Μαθηματικό μοντέλο

Το πρόβλημα μοντελοποιήθηκε σαν ένα πρόβλημα δρομολόγησης δυο επιπέδων όπως αυτό αναφέρθηκε σε προηγούμενο κεφάλαιο. Στόχος είναι η ελαχιστοποίηση της συνολικής απόστασης που διανύεται.

Σύνολα:

S: Σύνολο θέσεων στάθμευσης (δορυφορικοί σταθμοί φορτηγών).

C: Σύνολο πελατών που πρέπει να εξυπηρετηθούν.

A: Σύνολο τόξων (δρομολογίων).

A_1 : Τόξα που χρησιμοποιούνται από τα φορτηγά, συνδέοντας τις θέσεις στάθμευσης SS με την αποθήκη 00.

A_2 : Τόξα που χρησιμοποιούνται από τα drones, συνδέοντας τους πελάτες CC με τις θέσεις στάθμευσης SS.

k: Δείκτης που αντιπροσωπεύει τις υποδιαδρομές του φορτηγού.

s: Δείκτης που αντιπροσωπεύει τις υποδιαδρομές του drone.

Παράμετροι:

d_i : Ζήτηση του πελάτη i

Q: Χωρητικότητα του φορτηγού.

q: Χωρητικότητα του drone.

c_{ij} : Κόστος ταξιδιού μεταξύ των κόμβων i και j.

R_t : Μέγιστη εμβέλεια κάθε διαδρομής φορτηγού.

R_d : Μέγιστη εμβέλεια κάθε διαδρομής drone.

Μεταβλητές Απόφασης:

x_{ij}^k : Δυαδική μεταβλητή που παίρνει την τιμή 1 αν το φορτηγό εκτελεί την υποδιαδρομή k από τον κόμβο i στον κόμβο j, διαφορετικά 0.

y_{ij}^s : Δυαδική μεταβλητή που παίρνει την τιμή 1 αν το drone εκτελεί την υποδιαδρομή s από τον κόμβο i στον κόμβο j, διαφορετικά 0.

Ελαχιστοποίηση του συνολικού κόστους δρομολόγησης:

$$\min \sum_k \sum_{(i,j) \in A_1} c_{ij} x_{ij}^k + \sum_s \sum_{(i,j) \in A_2} c_{ij} y_{ij}^s$$

Το πρώτο άθροισμα αφορά το κόστος δρομολόγησης των φορτηγών. Το δεύτερο άθροισμα αφορά το κόστος δρομολόγησης

των drones. Το κόστος ορίζεται ως η συνολική απόσταση που διανύεται.

Περιορισμοί:

1. Ροή φορτηγών:

$$\sum_{(i,j) \in A_1} x_{ij}^k = \sum_{(j,i) \in A_1} x_{ji}^k \quad \forall k, \forall i \in S \cup \{0\}$$

Το φορτηγό που εισέρχεται σε έναν κόμβο πρέπει να εξέρχεται από αυτόν.

2. Εξυπηρέτηση θέσεων στάθμευσης:

$$\sum_k \sum_{(i,j) \in A_1} x_{ij}^k = 1 \quad \forall i \in S$$

Κάθε θέση στάθμευσης δέχεται ακριβώς μια επίσκεψη από το φορτηγό.

3. Ροή drones:

$$\sum_{(i,j) \in A_2} y_{ij}^s = \sum_{(j,i) \in A_2} y_{ji}^s \quad \forall s$$

Τα drones που αναχωρούν από μια θέση στάθμευσης επιστρέφουν στην ίδια. Κατά συνέπεια αυτό σημαίνει ότι προσγειώνονται στο ίδιο φορτηγό από το οποίο απογειώθηκαν.

4. Εξυπηρέτηση πελατών:

$$\sum_s \sum_{(i,j) \in A_2} y_{ij}^s = 1 \quad \forall j \in C$$

Κάθε πελάτης εξυπηρετείται ακριβώς μία φορά.

5. Χωρητικότητα φορτηγού:

$$\sum_{j \in S} d_j \cdot x_{ij}^k \leq Q \quad \forall k, \forall i \in S \cup \{0\}$$

Το φορτίο που μεταφέρεται από κάθε υποδιαδρομή φορτηγού δεν ξεπερνά τη χωρητικότητα του φορτηγού.

6. Χωρητικότητα drones:

$$\sum_{j \in C} d_j \cdot y_{ij}^s \leq q \quad \forall s, \forall i \in C$$

Το φορτίο που μεταφέρεται από κάθε υποδιαδρομή drone δεν ξεπερνά τη χωρητικότητα του drone.

7. Εμβέλεια φορτηγού:

$$\sum_{(i,j) \in A_1} d_{ij} \cdot x_{ij}^k \leq R_t \quad \forall k$$

Η συνολική απόσταση κάθε υποδιαδρομής φορτηγού πρέπει να είναι εντός της μέγιστης εμβέλειας.

8. Εμβέλεια drones:

$$\sum_{(i,j) \in A_2} d_{ij} \cdot y_{ij}^s \leq R_d \quad \forall s$$

Η συνολική απόσταση κάθε υποδιαδρομής drone πρέπει να είναι εντός της μέγιστης εμβέλειας.

9. Εξισορρόπηση φορτίων μεταξύ φορτηγών και θέσεων στάθμευσης:

$$\sum_k \sum_{(i,j) \in A_1} d_j \cdot x_{ij}^k = \sum_s \sum_{(i,j) \in A_2} d_j \cdot y_{ij}^s \quad \forall s \in S$$

Το φορτίο που παραδίδεται από τα φορτηγά στις θέσεις στάθμευσης πρέπει να είναι ίσο με το φορτίο που αναχωρεί από τις θέσεις στάθμευσης προς τους πελάτες.

7. Παρουσίαση κώδικα

Σε αυτό το κεφάλαιο θα εξηγηθεί επακριβώς η λειτουργία του υλοποιημένου αλγορίθμου μέσα από ψευδοκώδικα και διαγράμματα ροής. Ακόμα θα αναλυθεί σε ένα μικρό βαθμό η λογική και η διαδικασία ανάπτυξης του.

7.1 Μεθοδολογία επίλυσης

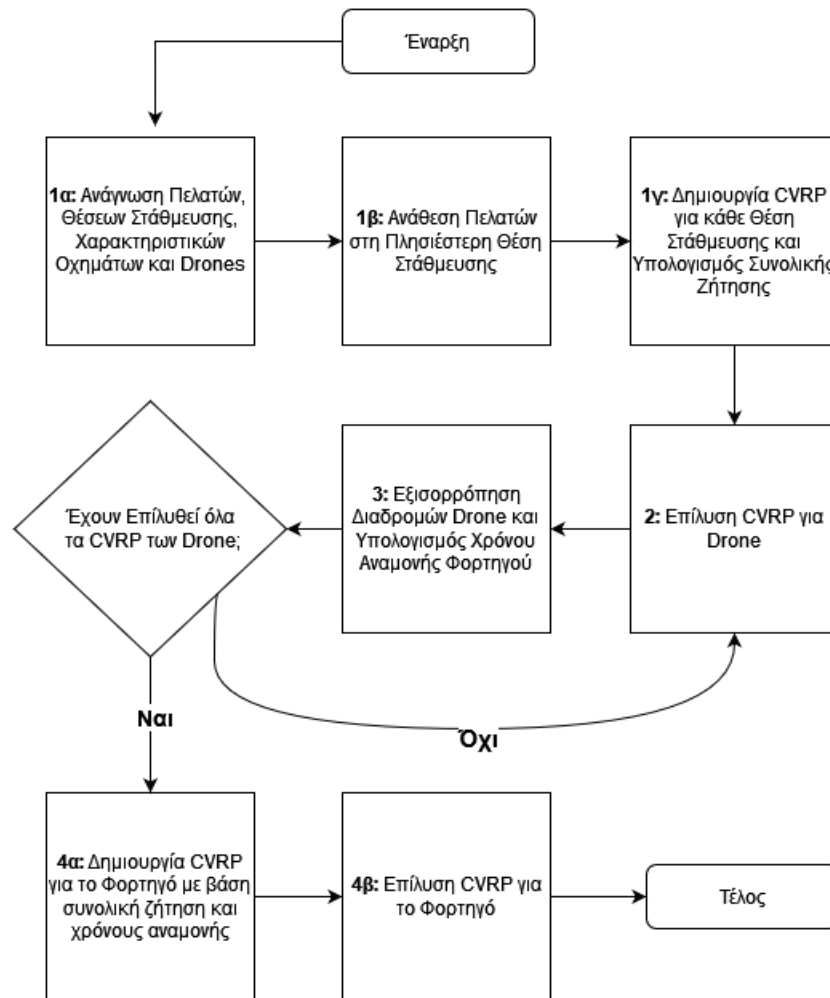
Το εν λόγω πρόβλημα δρομολόγησης χαρακτηρίζεται από υψηλή πολυπλοκότητα, καθώς περιλαμβάνει δύο διαφορετικούς τύπους οχημάτων, τα οποία κινούνται υπό διαφορετικούς περιορισμούς, ενώ επιπλέον το ένα μεταφέρει το άλλο. Μία πιθανή προσέγγιση θα μπορούσε να βασίζεται σε έναν αλγόριθμο που διαμορφώνει τη λύση βήμα-βήμα, μεταβαίνοντας μεταξύ διαφορετικών καταστάσεων (π.χ. κατάσταση: μετακίνηση φορτηγού από αποθήκη ή μετακίνηση drone από πελάτη σε πελάτη) και προσαρμόζοντας αντίστοιχα τους περιορισμούς. Η συγκεκριμένη μέθοδος εξετάστηκε, αλλά διαπιστώθηκε σύντομα ότι είτε απαιτείται μεγάλος αριθμός καταστάσεων είτε λιγότερες καταστάσεις με εξαιρετικά περίπλοκους κανόνες μετάβασης.

Επομένως, εξετάστηκε μια εναλλακτική στρατηγική. Διαπιστώθηκε ότι η δρομολόγηση των επίγειων και των εναέριων οχημάτων θα μπορούσε να προσεγγιστεί ξεχωριστά, επιτρέποντας την ανεξάρτητη βελτιστοποίηση για κάθε επίπεδο.

Η μεθοδολογία που παρουσιάζεται έχει τα εξής βήματα:

1. Οι πελάτες ομαδοποιούνται ανάλογα με την πλησιέστερη σε αυτούς θέση στάθμευσης. Η συνολική ζήτηση κάθε ομάδας υπολογίζεται και θεωρείται ως η ζήτηση της αντίστοιχης θέσης στάθμευσης.
2. Κάθε ομάδα πελατών, με την αντίστοιχη θέση στάθμευσης ως σημείο εκκίνησης, ορίζει ένα τοπικό πρόβλημα δρομολόγησης οχημάτων με περιορισμό χωρητικότητας και χρόνου. Αυτά τα προβλήματα επιλύονται εύκολα λόγω του περιορισμένου μεγέθους τους.
3. Μέσω ενός ευρετικού αλγορίθμου εξισορρόπησης, οι διαδρομές κατανέμονται κατάλληλα στον αριθμό των drones που μεταφέρει το φορτηγό. Υπολογίζεται, επίσης, ο χρόνος αναμονής του φορτηγού, με βάση τη χρονική διάρκεια της τελευταίας διαδρομής drone.
4. Αφού πλέον είναι γνωστές τόσο οι απαιτήσεις ζήτησης σε κάθε θέση στάθμευσης όσο και ο χρόνος αναμονής, διαμορφώνεται

και επιλύεται ένα τελικό πρόβλημα δρομολόγησης οχημάτων για το φορτηγό.



Εικόνα 7.1 Διάγραμμα ροής μεθοδολογίας επίλυσης

Με αυτόν τον τρόπο, ένα «δύσκολο» πρόβλημα δρομολόγησης μετατρέπεται σε πλήθος απλών προβλημάτων, καθιστώντας την επίλυσή του περισσότερο διαχειρίσιμη. Ωστόσο, ένας τέτοιος αλγόριθμος δεν μπορεί να εγγυηθεί την εύρεση της βέλτιστης λύσης, καθώς υπάρχουν περιπτώσεις όπου μια πιο αποδοτική κατανομή πελατών δεν θα εντοπιστεί.

Για παράδειγμα, σε μια οριακή περίπτωση, ένα drone από έναν πιο απομακρυσμένο σταθμό με επαρκή χωρητικότητα θα μπορούσε να εξυπηρετήσει έναν πελάτη που ανήκει σε ομάδα πιο κοντινού σταθμού. Αυτό θα ήταν επωφελές, ιδίως αν το drone του κοντινότερου σταθμού έχει ήδη εξαντλήσει τη χωρητικότητά του και θα απαιτούσε επιπλέον επιστροφές στον σταθμό.

Για τον λόγο αυτό, δοκιμάστηκε και μια παραλλαγή του αλγορίθμου, η οποία υλοποιεί τη διαδικασία επαναληπτικά. Συγκεκριμένα, οι πελάτες ανατίθενται με βάση την απόσταση, ενσωματώνοντας, παράλληλα, έναν παράγοντα τυχαιότητας. Επιπλέον, χρησιμοποιήθηκε ένας υποτυπώδης πίνακας φερομονών, ο οποίος επιτρέπει στον αλγόριθμο να «θυμάται» καλές αναθέσεις από προηγούμενες επαναλήψεις.

Μάλιστα αλγόριθμος Αποικιών Μυρμηγκιών (ACO) που υλοποιήθηκε είχε επίσης τη δυνατότητα να λαμβάνει ως είσοδο έναν πίνακα φερομόνων από προηγούμενες εκτελέσεις, ώστε να διατηρείται η σωρευμένη πληροφορία από προηγούμενα δρομολόγια. Παρόλα αυτά, διαπιστώθηκε ότι αυτή η μέθοδος ήταν εξαιρετικά χρονοβόρα χωρίς απαραίτητα να προσφέρει ουσιαστική βελτίωση στην ποιότητα των λύσεων και ως εκ τούτου δεν επιλέχθηκε.

7.2 Φιλοσοφία ανάπτυξης

Ο κώδικας αναπτύχθηκε με κύριο στόχο να είναι αναγνώσιμος, επεκτάσιμος, παραμετροποιήσιμος και ευέλικτος, ενώ παράλληλα σχεδιάστηκε ώστε να διευκολύνει τη διαδικασία ελέγχου. Σε γενικές γραμμές, η ιδανική ανάπτυξη λογισμικού περιλαμβάνει πρόσθετες απαιτήσεις, όπως η υψηλή ταχύτητα εκτέλεσης. Ωστόσο, στο πλαίσιο αυτής της διπλωματικής εργασίας, δόθηκε προτεραιότητα στα χαρακτηριστικά που εξυπηρετούν τη γρήγορη προτυποποίηση.

Αυτή η επιλογή κρίθηκε εύλογη, καθώς η ανάπτυξη μιας διπλωματικής εργασίας αποτελεί μια μη γραμμική διαδικασία, η οποία επωφελείται ιδιαίτερα από την ευελιξία και την προσαρμοστικότητα κατά τη διάρκεια της υλοποίησης.

Στη συνέχεια, θα παρουσιαστεί αναλυτικότερα η λειτουργία του αλγορίθμου, δομημένη σε μορφή ψευδοκώδικα. Κάθε τμήμα του ψευδοκώδικα αντιστοιχεί σε ένα συγκεκριμένο Python script, καθώς το λογισμικό αναπτύχθηκε σε περιβάλλον Python 3.

7.3 Problem Maker

Αυτό το τμήμα του κώδικα είναι υπεύθυνο για τη δημιουργία κατάλληλων προβλημάτων με σκοπό τη δοκιμή του αλγορίθμου. Αρχικά, ορίζονται οι παράμετροι του προβλήματος, όπως το μέγεθος του grid και τα διάφορα

χαρακτηριστικά των οχημάτων. Οι τιμές αυτών των παραμέτρων έχουν οριστεί αυθαίρετα, αλλά βρίσκονται εντός ρεαλιστικών πλαισίων.

1. `IMPORT` necessary libraries and modules
2. `#` Define parameters for problem creation
3. `SET` `grid_size` to 100
4. `SET` `num_clients` to 100
5. `SET` `num_parking_spots` to 20
6. `SET` `number_of_drones` to 4
7. `SET` `drones_consumption_rate` to 20
8. `SET` `drones_capacity` to 10
9. `SET` `drones_speed` to 50
10. `SET` `drones_max_route_time` to 20
11. `SET` `truck_capacity` to 500
12. `SET` `truck_speed` to 30
13. `SET` `truck_max_route_time` to 300
14. `SET` `truck_consumption_rate` to 800

Οι διάφορες παράμετροι αποθηκεύονται σαν αρχείο JSON. Οι συντεταγμένες των πελατών και των θέσεων στάθμευσης είναι τυχαίες.

15. `FUNCTION` `create_parameters_file(folder_path):`
16. Create a dictionary with all parameters
17. Save the dictionary as a JSON file in the specified folder
18. `FUNCTION` `generate_clients(grid_size, num_clients):`
19. Generate random client coordinates within the grid
20. RETURN client coordinates as a DataFrame
21. `FUNCTION` `generate_parking_spots(grid_size, num_parking_spots):`
22. Initialize a list for parking coordinates
23. Set the first parking spot to the center of the grid (depot)
24. Generate random coordinates for other parking spots
25. RETURN parking coordinates as a DataFrame
26. `FUNCTION` `generate_demands(num_clients):`
27. Generate random demands for clients
28. RETURN demands as a DataFrame

Οι θέσεις των πελατών και των σταθμών στάθμευσης αποθηκεύονται σε αρχεία Excel.

29. FUNCTION save_to_excel(folder_path, clients, parking_spots, demands):
30. Save client, parking spot, and demand data to separate Excel files in the specified folder

Επιπλέον, έχει υλοποιηθεί μια συνάρτηση που δημιουργεί μια οπτική αναπαράσταση του προβλήματος. Απεικονίζει τις συντεταγμένες των πελατών και των θέσεων στάθμευσης μέσα στο grid, χρησιμοποιώντας διαφορετικά χρώματα και σχήματα για κάθε κατηγορία.

31. FUNCTION visualize_problem(clients, parking_spots, grid_size, folder_path):
32. Create a plot of client and parking spot locations
33. Plot clients as red dots
34. Plot parking spots as green squares
35. Plot the depot as a black circle
36. Save the visualization as an image in the specified folder
37. Show the plot

Τέλος όλες οι συναρτήσεις καλούνται από την κεντρική.

38. FUNCTION main():
39. Open folder selection dialog
40. IF folder selected:
41. Create parameters file in the folder
42. Generate client data
43. Generate parking spot data
44. Generate demand data
45. Save all data to Excel files
46. Visualize the problem
47. ELSE:
48. Print message indicating no folder was selected

49. IF script is executed:
50. Run main()

7.4 Classes

Σε αυτό το τμήμα του κώδικα ορίζονται ορισμένες κλάσεις, οι οποίες χρησιμοποιούνται ως μεταβλητές σε άλλες συναρτήσεις του αλγορίθμου ACO. Οι κλάσεις αυτές έχουν σχεδιαστεί για να βελτιώσουν την επεκτασιμότητα και την αναγνωσιμότητα του κώδικα. Ουσιαστικά, πρόκειται για δομές δεδομένων που εμπλουτίζονται με πρόσθετες λειτουργικότητες, όπως ενσωματωμένες συναρτήσεις, επιτρέποντας την ευκολότερη διαχείριση των δεδομένων.

Στη συνέχεια, έχει υλοποιηθεί μια κλάση που αποθηκεύει όλες τις παραμέτρους που απαιτούνται για τη λειτουργία του αλγορίθμου ACO.

1. CLASS CVrpProblem:
2. FUNCTION __init__(client_positions, depot, demands, truck_capacity, max_route_time, consumption_rate, speed):
3. # Initialize the CVRP problem
4. Combine depot and client positions
5. Store problem constraints: truck capacity, route time, consumption rate, and speed
6. Calculate distances between all locations for routing

Έπειτα υπάρχει μια κλάση που αποθηκεύει όλες τις παραμέτρους που χρειάζεται ο Aco

7. CLASS AcoParameters:
8. FUNCTION __init__(n_ants, n_best, n_iterations, decay, alpha=1, beta=2):
9. # Initialize parameters for the Ant Colony Optimization (ACO) algorithm
10. - Set the number of ants, best-performing ants, iterations, and pheromone decay
11. - Configure ACO-specific parameters for balancing exploration and exploitation

Ακόμα δημιουργείται και μια κλάση που ορίζει ένα αντικείμενου τύπου μυρμήγκι. Αυτό αποθηκεύει την διαδρομή που έχει πραγματοποιήσει το μυρμήγκι ως τώρα, και περιλαμβάνει συναρτήσεις για την εύκολη αποθήκευση όλων των μεταβλητών του, όπως το κόστος του και το τρέχον επίπεδο χωρητικότητας.

12. CLASS Ant:
13. FUNCTION __init__():
14. # Initialize an ant for the algorithm
15. - Start its route at the depot
16. - Initialize route cost as zero
17. - Create an empty list to track truck capacity levels
- 18.
19. FUNCTION add_client(client):
20. # Add a client to the ant's route
- 21.
22. FUNCTION set_cost(cost):


```

23.      # Set the cost of the route
24.
25.  FUNCTION add_load(cap_level):
26.      # Log the truck's capacity level after visiting a client

```

7.5 Grouper_3

Αυτή η ενότητα του κώδικα υλοποιεί το πρώτο βήμα της μεθοδολογίας, διασπώντας το μεγάλο πρόβλημα σε πολλά μικρότερα αντικείμενα τύπου `CnvrProblem`. Αρχικά, ορίζεται μια κλάση που περιλαμβάνει όλα τα χαρακτηριστικά του συνολικού προβλήματος δρομολόγησης, καλύπτοντας τόσο τα επίγεια όσο και τα εναέρια οχήματα.

```

1. CLASS Problem:
2.   FUNCTION __init__(clients, parking_spots, depot, demands,
   number_of_drones, drones_consumption_rate,
3.       drones_capacity, drones_speed, drones_max_route_time,
   truck_capacity, truck_speed,
4.       truck_max_route_time, truck_consumption_rate):
5.       Initialize clients
6.       Initialize parking_spots
7.       Initialize depot
8.       Initialize demands
9.       Initialize client_assignments as empty
10.      Initialize number_of_drones
11.      Initialize drones_consumption_rate
12.      Initialize drones_capacity
13.      Initialize drones_speed
14.      Initialize drones_max_route_time
15.      Initialize truck_capacity
16.      Initialize truck_speed
17.      Initialize truck_max_route_time
18.      Initialize truck_consumption_rate

```

Ακόμα, ορίζεται μια συνάρτηση που εξάγει τα τελικά απλά προβλήματα σε μορφή Excel. Το συγκεκριμένο ενδιαμέσο βήμα έχει σχεδιαστεί για να διευκολύνει τον έλεγχο των υποπροβλημάτων απευθείας μέσα από το Excel.

```

19.  FUNCTION export_problem_files(folder_path):
20.      FOR each parking spot:
21.          Prepare data for clients assigned to the parking spot

```

22. Save data to an Excel file in the specified folder
--

Σε αυτό το σημείο πραγματοποιείται η ανάθεση των πελατών στις θέσεις στάθμευσης. Για κάθε πελάτη, υπολογίζονται όλες οι αποστάσεις από κάθε διαθέσιμη θέση στάθμευσης, και στη συνέχεια επιλέγεται η πλησιέστερη θέση. Η μέθοδος αυτή είναι απλή και αποτελεσματική και σε αντίθεση με άλλες πιο πολύπλοκες μεθόδους εξασφαλίζει την εγγύτητα των εναερίων οχημάτων με τους πελάτες καθώς, όπως θα εξηγηθεί και στην συνέχεια, είναι πολύ εύκολο οι πελάτες να βρεθούν εκτός εμβέλειας.

23. FUNCTION assign_clients_to_parking(problem, cost, iteration, pheromone_matrix):

24. IF iteration is 0:

25. Initialize client_assignments as empty list

26. FOR each client:

27. Calculate distances to all parking spots
--

28. Assign client to the nearest parking spot

29. Convert client_assignments to a numpy array

30. RETURN client_assignments

31. ELSE:

32. Initialize client_assignments as empty list

33. FOR each client:

34. Calculate distances to all parking spots
--

35. Assign client based on pheromone_matrix

36. RETURN client_assignments

Η συγκεκριμένη συνάρτηση παρέχει βασικά γραφικά στοιχεία διεπαφής με τον χρήστη που διευκολύνουν την επιλογή αρχείων.

37. FUNCTION file_opener():

38. Open folder selection dialog

39. Identify files for clients, parking spots, and demands
--

40. Load data from selected files

41. Separate depot from parking spots

42. IF demand file exists:

43. Load demands from the file

44. ELSE:

45. Generate random demands

46. Read parameters from JSON file

47. Initialize Problem object with the loaded data
--

48. RETURN Problem object

Σε αυτό το σημείο, οι πελάτες και οι θέσεις στάθμευσης που ανήκουν στο ίδιο υπό-πρόβλημα απεικονίζονται σε γραφική παράσταση με το ίδιο χρώμα για να ελεγχθεί η διαδικασία ομαδοποίησης.

```
49. FUNCTION plot_grouping(parking_spots, clients, problem, depot,
client_assignments):
50.   Create a plot for visualizing client assignments
51.   Add the depot as a black circle
52.   FOR each parking spot:
53.     Assign a unique color
54.     Plot the parking spot and its assigned clients
55.   Add legend and labels to the plot
56.   Display the plot
```

Η κεντρική συνάρτηση grouper_main καλεί όλες τις υπόλοιπες.

```
57. FUNCTION grouper_main(vrp, cost, iteration, pherormone_matrix):
58.   Print iteration number and cost
59.   Print parking spots
60.   Assign clients to parking spots using assign_clients_to_parking
61.   Update vrp client assignments
62.   Plot the grouping
63.   Export problem files for each subproblem
```

Τέλος υλοποιήθηκε και μια συνάρτηση για να μπορούν να διαβαστούν και προβλήματα σε μορφή αρχείων .dat.

```
64. FUNCTION convert_dat_to_excel():
65.   Open file selection dialog for a `.dat` file
66.   Read the file line by line
67.   Parse NODE_COORD_SECTION, SATELLITE_SECTION, and
DEMAND_SECTION
68.   Extract depot, satellite, and client coordinates
69.   Save data to Excel files for clients, parking spots, and demands
70.   Show the save folder in a file dialog
71.   Save the parsed data as Excel files
```

7.6 Multi aco 2

Η πρώτη συνάρτηση , `process subproblems` επιλύει τα προβλήματα δρομολόγησης διαδοχικά καλώντας τον αλγόριθμο ACO, υπολογίζει το κόστος τους, τον χρόνο που κάνουν να ολοκληρωθούν και δημιουργεί τις γραφικές παραστάσεις. Τέλος επιλύει το πρόβλημα δρομολόγησης των φορτηγών και εμφανίζει γραφικά όλες τις λύσεις μαζί.

```

1. IMPORT necessary libraries and modules
2. FUNCTION process_sub_problems(folder_path, depot_, Problem):
3.   Initialize L as number_of_drones
4.   Initialize c_rate as drones_consumption_rate
5.   Initialize c_rate_truck as truck_consumption_rate
6.   Initialize color list for visualizations
7.   Initialize color_index as 0
8.   Initialize plot
9.   Initialize lists for parking spots, demands, routing times, costs, routes,
   and wait times
10.  FOR each file in folder_path:
11.    IF file is a subproblem Excel file:
12.      Read problem data from the file
13.      Extract coordinates and demands
14.      Initialize depot from coordinates
15.      Create a CVrpProblem instance
16.      Initialize AcoForced colony with problem
17.      Run the colony to get the best route, cost, and wait time
18.      Append results to corresponding lists
19.      Plot clients, routes, and parking spots
20.  Plot truck routing:
21.    Add depot to parking spots
22.    Create a truck-level CVrpProblem instance
23.    Initialize and run AcoForced colony for the truck
24.    Plot the truck route
25.  RETURN total cost and total wait time

```

Η `file opener()` φροντίζει για την φόρτωση των δεδομένων στην κατάλληλη μορφή και παράγει τυχαίες απαιτήσεις πελατών αν αυτές υπολείπονται.

```

26. FUNCTION file_opener():
27.  Open folder selection dialog
28.  Identify files for clients, parking spots, and demands
29.  Load data from selected files
30.  Separate depot from parking spots

```

31. Generate random demands if no demand file exists
32. Read parameters from JSON
33. Initialize and RETURN a Problem object

Η main() είναι η πιο κεντρική συνάρτηση όλου του αλγορίθμου, αυτή η οποία «τρέχει» πρώτη και καλεί όλες τις υπόλοιπες.

34. FUNCTION main():
35. Open folder selection dialog to load files
36. Initialize depot and problem object
37. Set epochs, cost, and other initial variables
38. FOR each epoch:
39. Run grouper_main to group clients by parking spots
40. Run process_sub_problems to solve subproblems
41. Print total cost and time for the iteration
42. IF script is executed:
43. Run main()

7.7 AcoForced

Σε αυτό το τμήμα παρουσιάζεται η υλοποίηση του αλγορίθμου αποικίας μυρμηγκιών. Πιο συγκεκριμένα πρόκειται για μια υλοποίηση του Aco στην οποία τα η καλύτερα μυρμηγκία επηρεάζουν την ανανέωση του πίνακα φερομόνης. Ακόμα οι περιορισμοί εφαρμόζονται σε διάφορα σημεία του αλγορίθμου, και δεν υπολογίζονται σαν κόστος όποτε παραβιάζονται (συνάρτηση τιμωρίας). Αυτή η προσέγγιση είναι λιγότερο «ευέλικτη» από την χρήση συνάρτησης τιμωρίας, αλλά μιας και όλες οι λύσεις που παράγονται βρίσκονται στον χώρο των εφικτών, προσεγγίζεται η βέλτιστη πιο γρήγορα.

Στην αρχή αρχικοποιούνται ή φορτώνονται κάποιες βασικές μεταβλητές. Επιπρόσθετα δίνεται η δυνατότητα να φορτωθεί ο πίνακας φερομόνης ως πληροφορία από κάποια προηγούμενη επανάληψη. Αν όχι αρχικοποιείται αυτόματα.

1. CLASS AcoForced:
2. FUNCTION __init__(problem, parameters, pheromone_matrix, L):
3. Initialize L
4. Initialize problem
5. Initialize parameters
6. IF pheromone_matrix IS NOT None:

7. Initialize pheromone with pheromone_matrix
8. ELSE:
9. Initialize pheromone with ones divided by distance_matrix length
10. Initialize all_inds as range of distance_matrix length

Η συνάρτηση run() καλεί τις υπόλοιπες μεθόδους της κλάσης. «Κατασκευάζει» μυρμήγκια, τα ταξινομεί με βάση το κόστος και εφαρμόζει στα καλύτερα τοπική αναζήτηση. Στην συνέχεια ανανεώνει το πίνακα φερορμόνης και αποθηκεύει την ως τώρα καλύτερη λύση. Όταν το μέγιστο όριο επαναλήψεων πραγματοποιηθεί, υπολογίζει το κόστος της διαδρομής σε ενέργεια, σε χρόνο και αν υπάρχουν παραπάνω από ένα οχήματα, υπολογίζει και τον χρόνο ολοκλήρωσης μέχρι να επιστρέψει και τον τελευταίο όχημα στην αποθήκη (στην περίπτωση των UAVs που επιστρέφουν στο φορτηγό δηλαδή).

11. FUNCTION run():
12. Initialize best_path as None
13. Initialize best_distance as infinity
14. FOR i IN range of n_iterations:
15. Create list of ants using make_ant()
16. Sort ants by cost and select n_best
17. FOR each ant in best_ants:
18. Perform random_swap on ant
19. Perform two_opt_in_segments_between_returns on ant
20. Perform one_one_in_segments_between_returns on ant
21. Spread pheromone using best_ants
22. Multiply pheromone by decay
23. IF cost of best ant is less than best_distance:
24. Update best_path
25. Update best_distance
26. IF L > 1:
27. Calculate wait_time using calculate_wait_time()
28. ELSE:
29. Set wait time to best distance
30. Divide wait_time by speed
31. RETURN best_path, best_distance multiplied by consumption_rate, wait_time

Η συνάρτηση make_ant() κάθε φορά που καλείται κατασκευάζει μια λύση, δηλαδή ένα αντικείμενο τύπου μυρμήγκι. Εκτελείται προσθέτοντας σε κάθε επανάληψη έναν κόμβο μέχρι να χρησιμοποιηθούν όλοι. Κάθε φορά, ανανεώνει το κόστος, τον χρόνο και το φορτίο και ελέγχει αν η

χωρητικότητα του οχήματος ή μέγιστος χρόνος διαδρομής έχουν παραβιαστεί. Όποτε κριθεί απαραίτητο προσθέτει μια επιστροφή στην αποθήκη όπου το φορτίο και μέγιστος χρόνος μηδενίζονται ξανά. Ουσιαστικά εδώ εφαρμόζεται ο περιορισμός χωρητικότητας και χρόνου

```
32. FUNCTION make_ant():
33.   Initialize visited with depot
34.   Create new Ant
35.   Initialize capacity as truck_capacity
36.   Initialize current_route_time as 0
37.   WHILE visited length is less than num_clients:
38.     Calculate move_probabilities and unvisited
39.     Select move using pick_move()
40.     IF move IS NOT depot:
41.       Update capacity and current_route_time
42.       IF capacity is violated OR time exceeds max_route_time:
43.         Add depot to route
44.         Reset capacity and current_route_time
45.         Add client to route
46.     ELSE:
47.       Reset capacity and current_route_time
48.       Add move to visited
49.       Set ant cost using calculate_wait_time
50.       Add depot to route
51.   RETURN ant
```

Η συνάρτηση αυτή υπολογίζει το κόστος(σε μονάδες απόστασης).

```
52. FUNCTION calculate_cost(route):
53.   Initialize cost as 0
54.   FOR i IN range of route length minus 1:
55.     Add distance between route[i] and route[i + 1] to cost
56.   RETURN cost
```

Η calculate wait time() διασπά την καλύτερη λύση σε υπό-διαδρομές και υπολογίζει τον χρόνο αναμονής ως τον χρόνο που χρειάζεται το τελευταίο όχημα να ολοκληρώσει τις διαδρομές που του ανατέθηκαν.

```
57. FUNCTION calculate_wait_time(route, L):
58.   Split route into subroutes
59.   Filter subroutes with more than one client
60.   Calculate costs for each subroute
```

- | | |
|-----|---|
| 61. | Assign routes to drones using assign_routes() |
| 62. | RETURN wait_time |

Οι διάφορες διαδρομές ανατίθενται με ευρετικό τρόπο στα drones που είναι διαθέσιμα. Η κεντρική ιδέα είναι πολύ απλή. Αφού ο συνολικός χρόνος αναμονής είναι ο μεγαλύτερος χρόνος διαδρομών ενός UAV, αν οι διαδρομές χωριστούν έτσι ώστε να είναι εξισορροπημένες, η μέγιστη διάρκεια διαδρομής που θα προκύψει θα ελαχιστοποιηθεί.

Για να επιτευχθεί αυτό ακολουθείται η εξής στρατηγική. Αρχικοποιούνται κενές λίστες (μια για κάθε όχημα) και στην συνέχεια συμπληρώνονται επαναληπτικά. Σε κάθε επανάληψη, επιλέγεται η λίστα με τον μικρότερο συνολικό χρόνο, ώστε να αποφεύγεται η «υπερφόρτωση» κάποιας από αυτές. Η διαδικασία ξεκινά με την ανάθεση των μεγαλύτερων διαδρομών, ώστε να διατηρείται ισορροπία. Αντίθετα, αν η διαδικασία κατέληγε με τις μεγάλες διαδρομές, το σύστημα θα δυσκολευόταν να βρεθεί σε κατάσταση ισορροπίας, καθώς οι μεγάλες και απότομες αλλαγές στο τέλος της διαδικασίας θα οδηγούσαν σε ανισορροπία.

- | | |
|-----|--|
| 63. | FUNCTION assign_routes(subroute_costs, num_drones): |
| 64. | Sort subroute_costs in descending order |
| 65. | |
| 66. | Initialize drone_loads as a list of (0, drone_index) for all drones |
| 67. | Convert drone_loads into a heap |
| 68. | |
| 69. | Initialize assignments as an empty list for each drone |
| 70. | |
| 71. | FOR each cost in subroute_costs: |
| 72. | Pop the drone with the smallest current_load from drone_loads |
| 73. | Add cost to the assignments of the selected drone |
| 74. | Push the updated (current_load + cost, drone_index) back into the heap |
| 75. | |
| 76. | Calculate wait_time as the maximum current_load across all drones in drone_loads |
| 77. | |
| 78. | RETURN assignments, wait_time |

Η συνάρτηση calculate probabilities μεταβάλλει τις πιθανότητες έτσι ώστε να μηδενίζονται οι πιθανότητες επίσκεψης κόμβων που βρίσκονται ήδη στην τρέχουσα διαδρομή. Στην συνέχεια οι πιθανότητες κανονικοποιούνται. Ουσιαστικά εδώ εφαρμόζεται ο περιορισμός που ορίζει ότι κάθε πελάτης δέχεται μόνο μια επίσκεψη.


```

79.  FUNCTION calculate_probabilities(current, visited,
current_truck_capacity, current_route_time):
80.      Copy pheromone of current node
81.      Zero probabilities for visited nodes
82.      Get unvisited nodes
83.      Calculate distances from current node
84.      Compute probabilities using pheromone, alpha, beta
85.      Normalize probabilities
86.      RETURN probabilities, unvisited

```

Εδώ επιλέγεται ο επόμενος κόμβος που θα προστεθεί στην λύση. Ελέγχεται ακόμα αν υπάρχει κάποια κίνηση με μη μηδενική πιθανότητα, και αν δεν υπάρχει γίνεται επιστροφή στην αποθήκη.

```

87.  FUNCTION pick_move(move_probabilities):
88.      Replace NaNs with 0 in move_probabilities
89.      IF no valid moves:
90.          RETURN depot
91.      ELSE:
92.          RETURN weighted random choice based on probabilities

```

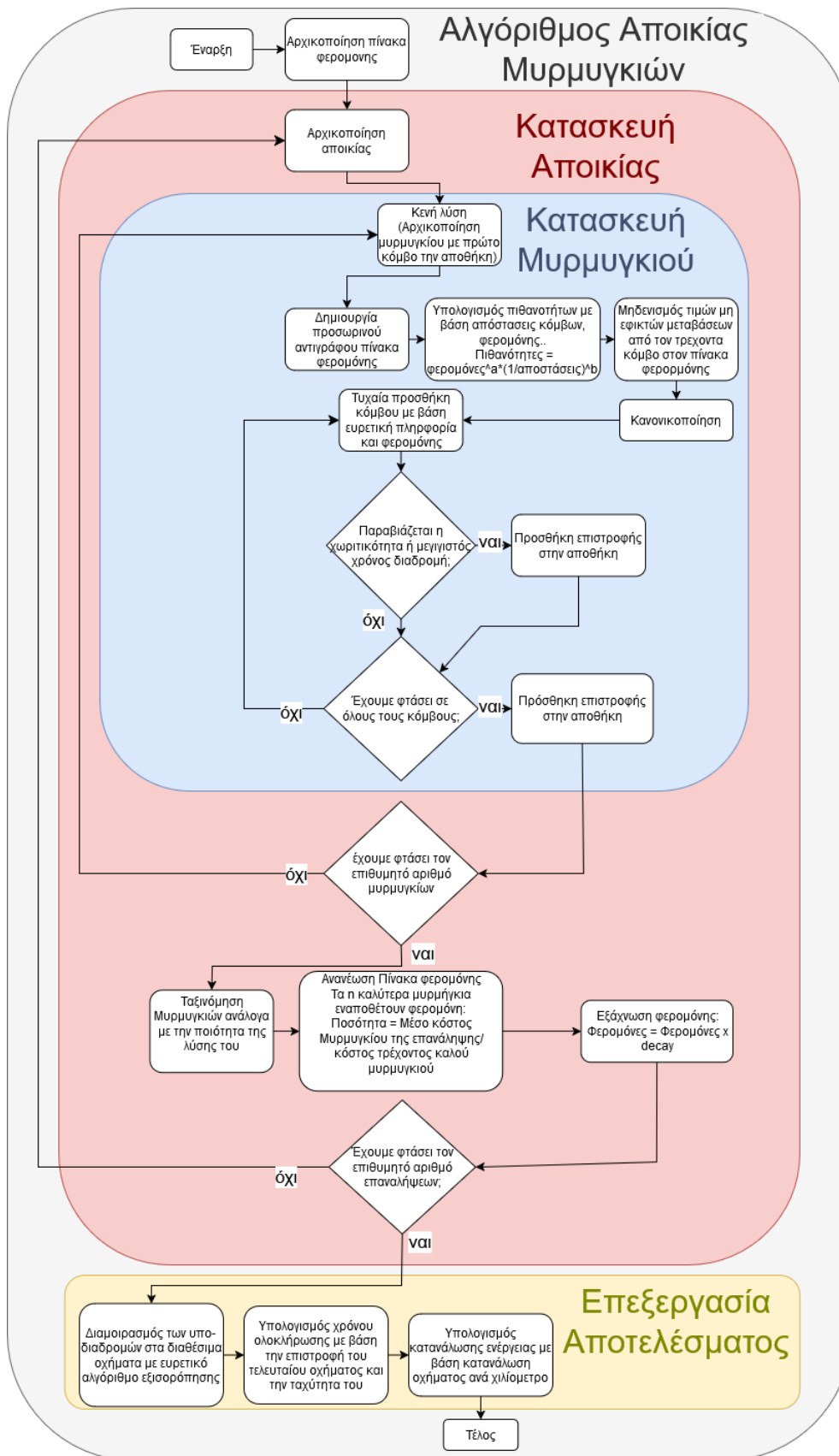
Όπως αναφέρθηκε παραπάνω η ανανέωση τη φερομόνης γίνεται με βάση τα η καλύτερα μυρμηγκία τα οποία εναποθέτουν φερομόνη ανάλογα με το κόστος τους σε σχέση με το μέσο κόστος όλων των μυρμηγκιών.

```

93.  FUNCTION spread_pheromone(best_ants):
94.      FOR each ant in best_ants:
95.          Update pheromone values for route pairs with adding
average_of_all_ant_costs/ant cost
96.          Normalize pheromone values

```

Επειδή το συγκεκριμένο κομμάτι του κώδικα που παρουσιάζεται είναι σχετικά πολύπλοκο, σχεδιάστηκε ένα διάγραμμα ροής που αναπαριστά αναλυτικά την λειτουργία του.



Εικόνα 7.2 Αναλυτικό διάγραμμα ροής υλοποιημένου αλγορίθμου αποικίας μυρμηγκιών

7.8 Local Search

Για την περεταίρω βελτιστοποίηση των διαδρομών χρησιμοποιείται τοπική αναζήτηση που τελειοποιεί ήδη αποδοτικές λύσεις. Εδώ υλοποιείται μια παραλλαγή τοπικής αναζήτησης one_one exchange (δυσό κόμβοι αλλάζουν τυχαία θέση). Ο αλγόριθμος μπορεί να πραγματοποιήσει ένα τυχαίο αριθμό ταυτόχρονων αλλαγών στην διαδρομή (συνήθως 2). Η διαδικασία αυτή μπορεί να επαναληφθεί έναν τυχαίο αριθμό φορών (συνήθως 30). Είναι σύνηθες να δοκιμάζονται πολλές ανταλλαγές καθώς αυτό υποστηρίζει την τελειοποίηση της λύσης. Η προσθήκη της δυνατότητας ταυτόχρονων αλλαγών είναι εξαιρετικά χρήσιμη για την διαφυγή από τοπικά ελάχιστα. Καθώς ο αλγόριθμος θα επιτρέψει κάποιες αλλαγές που στην αρχή οδηγούν σε «χειρότερες λύσεις» αλλά στην συνέχεια μπορούν να γίνουν μέρος μιας καλύτερης.

```
1. FUNCTION random_swap(ant, problem, exploration, exploitation):
2.   Initialize best_route with ant's route
3.   Initialize best_cost with ant's cost
4.   FOR iteration <= random number t [1 < t < exploitation]:
5.     FOR each random swap <= random number r [10 < r < exploitation]:
6.       Randomly select two indices i and j
7.       Create new_route by swapping route[i] and route[j]
8.       Calculate new_route_without_returns
9.       Create new_route with depot added where needed
10.      Calculate new_cost of new_route
11.      IF new_cost is less than best_cost:
12.        Update best_cost
13.        Update best_route
14.  RETURN best_route, best_cost
```

Επειδή η προηγούμενη συνάρτηση μπορεί να τοποθετήσει έναν πελάτη με μεγάλη ζήτηση σε μια υπό-διαδρομή στην οποία το όχημα να έχει ήδη υπερβεί το όριο χωρητικότητας του, οι επιστροφές στην αποθήκη χρειάζεται να εξεταστούν ξανά. Για αυτό φροντίζει η παρακάτω συνάρτηση.

```
15. FUNCTION put_returns_in_depot_when_needed(route_without_returns,
problem):
16.  Initialize capacity as truck_capacity
17.  Initialize unvisited with route_without_returns
18.  Initialize route as empty list
19.  WHILE unvisited is not empty:
20.    Get demand of the first client in unvisited
```

```

21.    IF demand exceeds capacity:
22.        Add depot to route
23.        Reset capacity
24.    Add client to route
25.    Remove client from unvisited
26.    Add depot to start and end of route
27.    RETURN route

```

Η συνάρτηση 2-opt εφαρμόζει τον τελεστή 2-opt.

```

28. FUNCTION two_opt(route, distance_matrix):
29.    Initialize best_route with route
30.    Initialize best_cost with cost of route
31.    FOR i IN range of route length minus 2:
32.        FOR j IN range of i + 1 to route length minus 1:
33.            Create new_route by reversing a segment of route
34.            Calculate new_cost of new_route
35.            IF new_cost is less than best_cost:
36.                Update best_cost
37.                Update best_route
38.    RETURN best_route, best_cost

```

Η two_opt_in_segments_between_returns εφαρμόζει τον τελεστή 2-opt όμως μόνο μέσα σε υπό-διαδρομές, «ξεμπλέκοντας» τις διαδρομές, χωρίς να παραβιάζει τους περιορισμούς.

```

39. FUNCTION two_opt_in_segments_between_returns(route, distance_matrix):
40.    Initialize Subroutes as empty list
41.    Split route into subroutes by depot visits
42.    Initialize final_route as empty list
43.    Initialize final_cost as 0
44.    FOR each subroute in Subroutes:
45.        Perform two_opt on subroute
46.        Add optimized subroute to final_route
47.        Add subroute cost to final_cost
48.    Add depot to end of final_route
49.    RETURN final_route, final_cost

```

Ακόμα η παρακάτω συνάρτηση εφαρμόζει τον τελεστή one_one στο ίδιο πνεύμα με την προηγούμενη

```

50. FUNCTION one_one_in_segments_between_returns(route, distance_matrix):

```

51. Initialize Subroutes as empty list
52. Split route into subroutes by depot visits
53. Initialize final_route as empty list
54. Initialize final_cost as 0
55. FOR each subroute in Subroutes:
56. Perform simple_one_one_exchange on subroute
57. Add optimized subroute to final_route
58. Add subroute cost to final_cost
59. Add depot to end of final_route
60. RETURN final_route, final_cost

8 Αποτελέσματα

Τα αποτελέσματα διακρίνονται σε δύο μέρη. Στο πρώτο μέρος επιλύονται έξι ομάδες προβλημάτων με διαφορετικές παραμέτρους της αποικίας μυρμηγκιών. Στόχος είναι να μελετηθεί η απόδοση του αλγορίθμου, και να προσδιοριστούν οι κατάλληλοι παράμετροι. Στο δεύτερο εξετάζονται διάφορα σενάρια πόλεων και οχημάτων και συγκρίνονται με ένα συμβατικό μοντέλο, ώστε να εξαχθούν συμπεράσματα για το ίδιο το συνεργατικό σύστημα διανομής.

8.1 Απόδοση αλγορίθμου

Για την δοκιμή του αλγορίθμου δημιουργήθηκαν τυχαία προβλήματα σε έξι μεγέθη. Για κάθε συνδυασμό αριθμού πελατών (C) και θέσεων στάθμευσης (P) δημιουργήθηκαν 3 παραλλαγές (a, b, c) για να εξασφαλιστεί ότι οι παρατηρήσεις που θα γίνουν πάνω στα αποτελέσματα δεν θα είναι βασισμένες σε διακυμάνσεις που προέκυψαν τυχαία. Αξίζει να σημειωθεί ότι οι θέσεις στάθμευσης δεν τοποθετήθηκαν στρατηγικά, αλλά τυχαία, ώστε να εξεταστεί η ικανότητα του αλγορίθμου να ομαδοποιεί τους πελάτες κατάλληλα και να προσαρμόζεται σε δυσμενείς συνθήκες.

Όνομα	Αριθμός Πελατών	Αριθμός Θέσεων Στάθμευσης	Ζήτηση σε kg
Small_C50_P5 _{a,b,c}	50	5	1-10
Small_C50_P10 _{a,b,c}	50	10	1-10
Medium_C100_P10 _{a,b,c}	100	10	1-10
Medium_C100_P20 _{a,b,c}	100	20	1-10
Large_C150_P15 _{a,b,c}	150	15	1-10
Large_C150_P30 _{a,b,c}	150	30	1-10

Πίνακας 8.1 Χαρακτηριστικά Προβλημάτων

Οι υπόλοιποι παράμετροι που έμειναν κοινοί για τα υπόλοιπα προβλήματα φαίνονται στον επόμενο πίνακα και αφορούν τα οχήματα.

Εναέρια οχήματα		Επίγεια οχήματα	
Μέση ταχύτητα	50 km/h	Μέση ταχύτητα	30Km/h
Εμβέλεια	100 km	Εμβέλεια	200 km
Χωρητικότητα	30 kg	Χωρητικότητα	800 kg
Αριθμός	3	Αριθμός	1
Κατανάλωση	Wh/km	Κατανάλωση	Wh/km

Πίνακας 8.2 Χαρακτηριστικά Οχημάτων

Σε αυτά τα προβλήματα δοκιμάστηκε ένα πλήθος συνδυασμών παραμέτρων. Γενικά Ο ACO, έχει πολλές παραμέτρους, και ακριβώς λόγω του πλήθους τους (στη συγκεκριμένη περίπτωση 6), θα ήταν εξαιρετικά δύσκολο να εξεταστούν οι συνδυασμοί όλων. Κατά συνέπεια επιλέχθηκε ένας συνδυασμός ως «βάση» και οι υπόλοιποι διαμορφώθηκαν σαν διαφοροποιήσεις από αυτήν. Οι συνδυασμοί που δοκιμάστηκαν είναι:

Base	15	5	15	0,5	1	2
A	20	5	15	0,5	1	2
B	30	5	15	0,5	1	2
C	15	10	15	0,5	1	2
D	20	10	15	0,5	1	2
E	30	15	15	0,5	1	2
F	15	5	20	0,5	1	2
G	15	5	30	0,5	1	2
H	15	5	40	0,5	1	2
I	20	10	30	0,5	1	2
J	15	5	15	0,7	1	2
K	15	5	15	0,9	1	2
L	15	5	15	0,5	2	2
M	15	5	15	0,5	3	2
O	15	5	15	0,5	1	1
P	15	5	15	0,5	1	3

Πίνακας 8.3 Συνδυασμοί Παραμέτρων

Στην γκρι γραμμή βρίσκεται ο συνδυασμός παραμέτρων που πάρθηκε ως βάση. Οι υπόλοιποι συνδυασμοί (A έως P) είναι διαφοροποιήσεις του. Οι τιμές με έντονα γράμματα αντιπροσωπεύουν τις αλλαγές από το σενάριο βάση.

8.1.1 Προβλήματα με 100 πελάτες και 5 θέσεις στάθμευσης

C100_P5a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,33	2791,05	2934,46	2832,90
A	20	5	15	0,5	1	2	6,42	2822,72	2887,82	2850,46
B	30	5	15	0,5	1	2	8,68	2810,28	2880,10	2848,05
C	15	10	15	0,5	1	2	6,52	2774,02	2882,15	2825,63
D	20	10	15	0,5	1	2	7,71	2786,33	2854,09	2817,02
E	30	15	15	0,5	1	2	11,24	2780,97	2891,63	2819,76
F	15	5	20	0,5	1	2	6,90	2781,42	2899,80	2844,37
G	15	5	30	0,5	1	2	10,00	2787,32	2881,30	2825,85
H	15	5	40	0,5	1	2	13,43	2784,50	2840,62	2816,83
I	20	10	30	0,5	1	2	14,79	2783,78	2868,46	2818,15
J	15	5	15	0,7	1	2	6,69	2793,90	2882,35	2825,38
K	15	5	15	0,9	1	2	5,42	2787,34	2932,78	2840,06
L	15	5	15	0,5	2	2	5,36	2833,86	2927,62	2870,44
M	15	5	15	0,5	3	2	5,48	2854,89	2970,42	2900,53
O	15	5	15	0,5	1	1	5,47	2813,57	2953,11	2872,50
P	15	5	15	0,5	1	3	5,69	2832,32	2882,11	2861,04
C100_P5b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,54	2882,52	2983,66	2938,03
A	20	5	15	0,5	1	2	6,85	2943,19	3004,95	2962,24
B	30	5	15	0,5	1	2	9,10	2887,75	3006,48	2955,69
C	15	10	15	0,5	1	2	6,68	2882,92	2969,76	2946,45
D	20	10	15	0,5	1	2	7,82	2874,79	2977,66	2936,71
E	30	15	15	0,5	1	2	11,47	2911,89	2967,46	2939,66
F	15	5	20	0,5	1	2	6,99	2909,56	3036,54	2951,13
G	15	5	30	0,5	1	2	10,16	2884,40	2962,06	2939,34
H	15	5	40	0,5	1	2	13,32	2912,37	2995,52	2934,45
I	20	10	30	0,5	1	2	14,73	2912,30	2982,75	2933,97
J	15	5	15	0,7	1	2	6,54	2897,00	3016,26	2944,68
K	15	5	15	0,9	1	2	5,32	2955,49	3029,64	2979,16
L	15	5	15	0,5	2	2	5,34	2925,54	3005,27	2961,65
M	15	5	15	0,5	3	2	5,44	2969,72	3098,00	3011,32
O	15	5	15	0,5	1	1	5,38	2928,58	3028,82	2985,69
P	15	5	15	0,5	1	3	5,41	2876,49	2969,38	2944,44
C100_P5c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,94	2226,75	2258,61	2241,51
A	20	5	15	0,5	1	2	7,13	2195,06	2303,35	2252,42
B	30	5	15	0,5	1	2	9,85	2228,97	2274,75	2249,27
C	15	10	15	0,5	1	2	7,55	2198,03	2261,47	2234,79
D	20	10	15	0,5	1	2	8,61	2201,44	2261,75	2233,09
E	30	15	15	0,5	1	2	11,96	2189,07	2243,00	2221,90
F	15	5	20	0,5	1	2	7,54	2196,81	2278,43	2239,97
G	15	5	30	0,5	1	2	10,92	2217,66	2261,02	2240,99
H	15	5	40	0,5	1	2	14,14	2183,00	2267,31	2232,24
I	20	10	30	0,5	1	2	15,66	2188,69	2255,49	2219,50
J	15	5	15	0,7	1	2	7,06	2178,92	2264,00	2219,02
K	15	5	15	0,9	1	2	5,82	2198,85	2271,98	2232,08
L	15	5	15	0,5	2	2	5,85	2234,94	2299,89	2257,47
M	15	5	15	0,5	3	2	6,19	2237,66	2304,19	2276,34
O	15	5	15	0,5	1	1	6,04	2186,80	2277,22	2214,16
P	15	5	15	0,5	1	3	6,00	2237,51	2296,10	2263,15

Πίνακας 8.4 Αποτελέσματα C100_P5

Οι έγχρωμες μπάρες αντιπροσωπεύουν τις ποσοστιαίες διαφορές των τιμών σε μια στήλη. Δηλαδή η μεγαλύτερη τιμή παρουσιάζει μια πλήρη μπάρα (100%), η μικρότερη κενό (0%) και οι υπόλοιπες τα ενδιάμεσα ποσοστά αυτών. Η κόκκινη γραμμή επισημαίνει την χειρότερη μέση λύση, η πράσινη την καλύτερη, και οι μπλε γραμμές είναι ενδεικτικές καλών συνδυασμών που μεγιστοποιούν την απόδοση χωρίς να αυξάνουν σημαντικά τον χρόνο εκτέλεσης. Στην συγκεκριμένη περίπτωση ο χειρότερος συνδυασμός είναι ο **M** ($\alpha = 3$). Ο **I** (ants = 20, best ants = 30) και ο **H** (epochs = 30) είναι πολύ αποδοτικοί. Αυτό είναι αναμενόμενο καθώς αυτοί οι συνδυασμοί

χρησιμοποιούν σημαντικά περισσότερα μυρμήγκια. Ιδιαίτερο ενδιαφέρον παρουσιάζουν επίσης ο **J** (decay = 0,7) και ο **O** (a = 1). Σε γενικές γραμμές η αύξηση στο decay και η μείωση του a, ενθαρρύνουν περισσότερο την εξερεύνηση, γιατί μειώνουν την επίδραση της φερομόνης.

8.1.2 Προβλήματα με 100 πελάτες και 10 θέσεις στάθμευσης:

C100_P10a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,16	2592,99	2659,74	2621,93
A	20	5	15	0,5	1	2	5,87	2592,68	2646,27	2622,74
B	30	5	15	0,5	1	2	7,87	2604,10	2664,12	2632,86
C	15	10	15	0,5	1	2	6,03	2556,98	2645,79	2601,15
D	20	10	15	0,5	1	2	7,03	2594,57	2640,93	2617,43
E	30	15	15	0,5	1	2	10,12	2589,74	2672,39	2627,86
F	15	5	20	0,5	1	2	6,28	2603,38	2676,75	2633,75
G	15	5	30	0,5	1	2	9,02	2554,26	2660,06	2615,60
H	15	5	40	0,5	1	2	11,80	2542,11	2645,47	2613,61
I	20	10	30	0,5	1	2	13,31	2577,25	2686,45	2620,41
J	15	5	15	0,7	1	2	6,07	2548,66	2626,27	2588,71
K	15	5	15	0,9	1	2	4,94	2605,63	2725,71	2645,70
L	15	5	15	0,5	2	2	4,92	2632,01	2712,65	2658,80
M	15	5	15	0,5	3	2	5,01	2652,92	2755,22	2682,43
O	15	5	15	0,5	1	1	4,92	2587,92	2691,79	2618,92
P	15	5	15	0,5	1	3	4,99	2628,89	2682,76	2652,01
C100_P10b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,93	2813,71	2905,76	2870,18
A	20	5	15	0,5	1	2	5,83	2864,21	2911,29	2883,45
B	30	5	15	0,5	1	2	7,77	2868,87	2932,67	2897,73
C	15	10	15	0,5	1	2	5,95	2820,77	2874,26	2852,46
D	20	10	15	0,5	1	2	6,91	2844,51	2911,92	2870,09
E	30	15	15	0,5	1	2	10,41	2860,13	2884,00	2874,04
F	15	5	20	0,5	1	2	6,37	2859,93	2905,88	2880,92
G	15	5	30	0,5	1	2	8,96	2851,33	2954,04	2883,18
H	15	5	40	0,5	1	2	11,75	2812,83	2919,47	2867,94
I	20	10	30	0,5	1	2	13,15	2813,04	2882,07	2852,65
J	15	5	15	0,7	1	2	5,98	2811,60	2898,23	2857,59
K	15	5	15	0,9	1	2	4,85	2846,02	2902,96	2873,01
L	15	5	15	0,5	2	2	4,87	2875,30	2984,12	2924,66
M	15	5	15	0,5	3	2	4,97	2873,97	2981,87	2930,42
O	15	5	15	0,5	1	1	4,86	2839,33	2895,70	2867,07
P	15	5	15	0,5	1	3	4,98	2865,58	2916,07	2888,86
C100_P10c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,90	2565,60	2604,64	2582,14
A	20	5	15	0,5	1	2	5,86	2577,73	2648,20	2603,77
B	30	5	15	0,5	1	2	7,80	2597,72	2673,23	2629,53
C	15	10	15	0,5	1	2	6,02	2528,42	2626,75	2572,27
D	20	10	15	0,5	1	2	7,06	2544,19	2607,43	2568,70
E	30	15	15	0,5	1	2	10,19	2558,25	2602,11	2581,22
F	15	5	20	0,5	1	2	6,26	2557,53	2669,48	2592,12
G	15	5	30	0,5	1	2	8,97	2549,43	2636,48	2594,66
H	15	5	40	0,5	1	2	11,70	2546,40	2622,57	2587,87
I	20	10	30	0,5	1	2	13,27	2541,60	2599,25	2567,86
J	15	5	15	0,7	1	2	6,05	2539,22	2597,55	2572,24
K	15	5	15	0,9	1	2	4,91	2538,40	2603,71	2574,09
L	15	5	15	0,5	2	2	4,88	2572,60	2664,27	2611,66
M	15	5	15	0,5	3	2	4,95	2589,60	2691,86	2645,39
O	15	5	15	0,5	1	1	4,88	2526,97	2604,33	2563,08
P	15	5	15	0,5	1	3	4,99	2576,90	2639,68	2607,59

Πίνακας 8.5 Αποτελέσματα C100_P10

Από τα παραπάνω δεδομένα διαπιστώνεται ότι η αύξηση των θέσεων στάθμευσης από 5 σε 10 φαίνεται να επιδεινώνει την απόδοση καθώς ο μέσος όρος των καλύτερων είναι για τα C100_P5 είναι **2576 km**, ενώ για τα C100_P10 είναι **2649, 58 km**. Ωστόσο η διαφορά είναι μικρή για να εξαχθούν συμπεράσματα, λαμβάνοντας επίσης υπ' όψιν ότι τα προβλήματα έχουν

δημιουργηθεί τυχαία, (δηλαδή πχ το C100_P5a και C100_P10a, δεν έχουν κοινές θέσεις πελατών). Επίσης παρατηρείται ότι ο **C** (best ants = 10) αναδύεται σαν δυνατός συνδυασμός. Παράλληλα ο **M** (αυξημένο a = 3) παραμένει ο χειρότερος, ενώ οι **J** (αυξημένο decay = 0,7) και **O** (μειωμένο a = 1) παραμένουν στους καλύτερους.

8.1.3 Προβλήματα με 150 πελάτες και 5 θέσεις στάθμευσης:

C150_P5a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,32	2201,02	2251,12	2227,36
A	20	5	15	0,5	1	2	5,27	2187,68	2240,42	2211,36
B	30	5	15	0,5	1	2	7,52	2184,99	2240,01	2210,21
C	15	10	15	0,5	1	2	4,86	2182,85	2258,82	2222,33
D	20	10	15	0,5	1	2	5,91	2176,55	2248,62	2211,67
E	30	15	15	0,5	1	2	8,73	2174,34	2253,76	2214,53
F	15	5	20	0,5	1	2	5,49	2175,69	2231,96	2203,69
G	15	5	30	0,5	1	2	8,54	2167,16	2228,87	2193,98
H	15	5	40	0,5	1	2	10,88	2149,18	2215,71	2184,80
I	20	10	30	0,5	1	2	11,55	2152,51	2221,70	2177,39
J	15	5	15	0,7	1	2	4,81	2198,25	2237,83	2214,84
K	15	5	15	0,9	1	2	4,24	2181,74	2256,92	2222,56
L	15	5	15	0,5	2	2	4,18	2191,43	2316,20	2241,11
M	15	5	15	0,5	3	2	4,27	2215,99	2335,66	2272,33
O	15	5	15	0,5	1	1	4,16	2245,91	2323,78	2290,45
P	15	5	15	0,5	1	3	4,62	2171,65	2224,66	2190,18
C150_P5b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,47	2354,92	2443,29	2407,36
A	20	5	15	0,5	1	2	5,42	2392,32	2448,02	2422,34
B	30	5	15	0,5	1	2	7,67	2378,85	2436,60	2407,90
C	15	10	15	0,5	1	2	4,94	2373,09	2441,97	2409,50
D	20	10	15	0,5	1	2	6,09	2385,83	2438,36	2413,19
E	30	15	15	0,5	1	2	9,00	2356,38	2433,87	2394,47
F	15	5	20	0,5	1	2	5,68	2360,96	2435,56	2404,15
G	15	5	30	0,5	1	2	8,35	2368,72	2433,27	2395,06
H	15	5	40	0,5	1	2	11,08	2347,64	2400,98	2376,90
I	20	10	30	0,5	1	2	11,72	2322,60	2428,96	2387,32
J	15	5	15	0,7	1	2	4,81	2373,08	2454,69	2415,98
K	15	5	15	0,9	1	2	4,19	2387,61	2481,00	2415,51
L	15	5	15	0,5	2	2	4,17	2388,42	2572,30	2452,88
M	15	5	15	0,5	3	2	4,28	2439,62	2525,52	2472,18
O	15	5	15	0,5	1	1	4,19	2410,85	2526,72	2483,89
P	15	5	15	0,5	1	3	4,31	2344,81	2449,77	2411,94
C150_P5c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,25	2575,66	2640,17	2610,44
A	20	5	15	0,5	1	2	5,33	2577,63	2665,45	2611,52
B	30	5	15	0,5	1	2	7,69	2576,55	2671,41	2618,30
C	15	10	15	0,5	1	2	4,73	2566,98	2631,80	2600,16
D	20	10	15	0,5	1	2	5,79	2562,10	2633,07	2597,16
E	30	15	15	0,5	1	2	8,57	2568,69	2620,39	2588,63
F	15	5	20	0,5	1	2	5,57	2546,61	2630,56	2589,26
G	15	5	30	0,5	1	2	8,03	2532,65	2615,53	2586,58
H	15	5	40	0,5	1	2	10,63	2530,02	2610,06	2571,69
I	20	10	30	0,5	1	2	11,55	2529,92	2599,39	2555,74
J	15	5	15	0,7	1	2	4,80	2488,02	2633,19	2590,23
K	15	5	15	0,9	1	2	4,18	2587,95	2698,30	2625,52
L	15	5	15	0,5	2	2	4,15	2569,02	2697,98	2649,19
M	15	5	15	0,5	3	2	4,28	2637,09	2763,62	2674,70
O	15	5	15	0,5	1	1	4,16	2663,33	2749,51	2696,20
P	15	5	15	0,5	1	3	4,32	2560,34	2639,40	2593,85

Πίνακας 8.6 Αποτελέσματα C150_P5

Η αύξηση των πελατών στους 150 αλλάζει αρκετά τα αποτελέσματα καθώς πλέον ο **O** (μειωμένο a = 1) είναι ο χειρότερος συνδυασμός. Οι συνδυασμοί **I** (ants = 20, best ants = 30), και **H** (αυξημένες επαναλήψεις = 40), και γενικά οι

συνδυασμοί με περισσότερα μυρμήγκια φαίνεται να αποδίδουν. Ενδεχομένως αυτό να οφείλεται στο ότι περισσότεροι πελάτες αυξάνουν την πολυπλοκότητα του προβλήματος και έτσι απαιτούνται περισσότεροι «πόροι». Αντίθετα ο **J** (αυξημένο decay = 0,7) δεν είναι τόσο αποδοτικός αλλά είναι ο μόνος που παρέχει σχετικά καλές λύσεις σε σύντομο χρόνο.

8.1.4 Προβλήματα με 150 πελάτες και 10 θέσεις στάθμευσης:

C150_P10a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,60	1960,46	2039,05	1996,67
A	20	5	15	0,5	1	2	5,26	1946,77	2026,53	1982,94
B	30	5	15	0,5	1	2	7,13	1972,91	2011,96	1992,50
C	15	10	15	0,5	1	2	4,99	1950,14	1999,57	1974,62
D	20	10	15	0,5	1	2	5,97	1928,90	2000,94	1977,47
E	30	15	15	0,5	1	2	8,71	1932,46	2007,14	1979,15
F	15	5	20	0,5	1	2	5,42	1977,54	2031,07	1995,13
G	15	5	30	0,5	1	2	7,91	1944,29	2014,07	1982,36
H	15	5	40	0,5	1	2	10,39	1943,47	1992,59	1971,71
I	20	10	30	0,5	1	2	11,40	1958,51	1991,45	1974,62
J	15	5	15	0,7	1	2	4,95	1937,04	2013,58	1981,09
K	15	5	15	0,9	1	2	4,16	1975,09	2010,30	1997,22
L	15	5	15	0,5	2	2	4,17	1933,69	2064,23	2008,47
M	15	5	15	0,5	3	2	4,28	1973,35	2079,37	2026,19
O	15	5	15	0,5	1	1	4,19	1972,31	2033,99	1999,40
P	15	5	15	0,5	1	3	4,27	1971,13	2034,27	2009,35
C150_P10b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,24	2123,96	2233,22	2176,17
A	20	5	15	0,5	1	2	5,09	2136,35	2212,89	2182,24
B	30	5	15	0,5	1	2	7,18	2161,56	2221,06	2192,79
C	15	10	15	0,5	1	2	4,77	2127,61	2218,53	2164,81
D	20	10	15	0,5	1	2	5,82	2139,99	2234,77	2179,23
E	30	15	15	0,5	1	2	8,76	2117,59	2215,55	2187,04
F	15	5	20	0,5	1	2	5,41	2101,34	2204,82	2178,79
G	15	5	30	0,5	1	2	7,85	2129,55	2201,56	2158,06
H	15	5	40	0,5	1	2	10,25	2109,89	2204,43	2150,97
I	20	10	30	0,5	1	2	11,40	2126,54	2193,66	2156,97
J	15	5	15	0,7	1	2	5,11	2102,11	2189,65	2154,00
K	15	5	15	0,9	1	2	4,32	2135,44	2214,19	2173,57
L	15	5	15	0,5	2	2	4,26	2157,22	2245,48	2186,61
M	15	5	15	0,5	3	2	4,28	2144,95	2232,39	2197,16
O	15	5	15	0,5	1	1	4,28	2087,59	2191,72	2150,07
P	15	5	15	0,5	1	3	4,29	2171,67	2206,86	2189,65
C150_P10c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,40	2408,47	2503,63	2457,78
A	20	5	15	0,5	1	2	5,31	2441,31	2484,36	2462,63
B	30	5	15	0,5	1	2	7,32	2430,06	2546,88	2478,75
C	15	10	15	0,5	1	2	5,15	2430,14	2498,41	2466,11
D	20	10	15	0,5	1	2	6,33	2417,35	2498,31	2446,87
E	30	15	15	0,5	1	2	8,98	2417,77	2490,26	2456,94
F	15	5	20	0,5	1	2	5,62	2383,09	2456,44	2421,00
G	15	5	30	0,5	1	2	8,22	2361,84	2464,06	2411,49
H	15	5	40	0,5	1	2	10,75	2380,70	2469,84	2416,94
I	20	10	30	0,5	1	2	11,84	2386,21	2449,30	2420,17
J	15	5	15	0,7	1	2	5,17	2425,22	2487,72	2449,61
K	15	5	15	0,9	1	2	4,31	2433,70	2524,57	2464,37
L	15	5	15	0,5	2	2	4,48	2414,34	2549,77	2487,46
M	15	5	15	0,5	3	2	4,45	2459,41	2547,02	2501,99
O	15	5	15	0,5	1	1	4,37	2384,25	2493,99	2438,37
P	15	5	15	0,5	1	3	4,53	2422,59	2504,83	2466,66

Πίνακας 8.7 Αποτελέσματα C100_P10

Με την αύξηση των θέσεων στάθμευσης σε 10, η συμπεριφορά του αλγορίθμου αρχίζει να θυμίζει περισσότερο στα προβλήματα με 100 πελάτες. Ουσιαστικά με την αύξηση των θέσεων στάθμευσης η πολυπλοκότητα μειώθηκε καθώς τα προβλήματα δρομολόγησης των drone μπορεί πολλαπλασιάστηκαν σε αριθμό αλλά συρρικνώθηκαν σε μέγεθος (περίπου

στο μισό). Ο χειρότερος συνδυασμός είναι πάλι ο **M**. Ο **O** και ο **J** εμφανίζουν ξανά καλή απόδοση, επειδή μάλλον στο συγκεκριμένο σενάριο ο αλγόριθμος μπορεί να επωφεληθεί περισσότερο από την εξερεύνηση. Τέλος πρέπει να παρατηρηθεί ότι οι αποστάσεις μειώνονται και στα τρία προβλήματα με την προσθήκη περισσότερων θέσεων στάθμευσης.

8.1.5 Προβλήματα με 200 πελάτες και 10 θέσεις στάθμευσης:

C200_P10a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,80	2692,56	2775,14	2734,22
A	20	5	15	0,5	1	2	6,89	2661,62	2745,40	2701,04
B	30	5	15	0,5	1	2	9,80	2658,22	2716,71	2689,35
C	15	10	15	0,5	1	2	6,40	2672,28	2767,67	2715,53
D	20	10	15	0,5	1	2	7,78	2631,86	2748,97	2697,81
E	30	15	15	0,5	1	2	11,33	2634,45	2733,49	2690,04
F	15	5	20	0,5	1	2	7,07	2682,62	2757,26	2717,12
G	15	5	30	0,5	1	2	10,85	2634,09	2715,02	2681,87
H	15	5	40	0,5	1	2	14,58	2625,91	2705,46	2664,14
I	20	10	30	0,5	1	2	15,47	2636,45	2703,76	2665,10
J	15	5	15	0,7	1	2	6,60	2656,30	2764,98	2693,25
K	15	5	15	0,9	1	2	5,36	2672,56	2778,83	2736,04
L	15	5	15	0,5	2	2	5,34	2716,06	2786,63	2741,83
M	15	5	15	0,5	3	2	5,57	2713,30	2816,29	2756,46
O	15	5	15	0,5	1	1	5,50	2727,69	2820,50	2755,51
P	15	5	15	0,5	1	3	5,72	2675,93	2739,95	2710,07

C200_P10b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,46	2440,78	2499,07	2473,90
A	20	5	15	0,5	1	2	6,60	2443,98	2528,05	2483,82
B	30	5	15	0,5	1	2	9,49	2463,36	2558,28	2494,05
C	15	10	15	0,5	1	2	6,35	2420,78	2524,16	2466,66
D	20	10	15	0,5	1	2	7,65	2444,40	2502,14	2469,84
E	30	15	15	0,5	1	2	11,02	2436,35	2516,29	2480,29
F	15	5	20	0,5	1	2	7,03	2412,74	2510,61	2462,17
G	15	5	30	0,5	1	2	10,16	2426,83	2498,85	2454,39
H	15	5	40	0,5	1	2	13,73	2405,09	2471,55	2433,54
I	20	10	30	0,5	1	2	14,80	2439,61	2493,54	2454,73
J	15	5	15	0,7	1	2	6,12	2449,45	2513,76	2477,81
K	15	5	15	0,9	1	2	5,24	2460,55	2529,69	2491,10
L	15	5	15	0,5	2	2	5,26	2480,78	2619,02	2529,76
M	15	5	15	0,5	3	2	5,34	2492,65	2640,32	2540,08
O	15	5	15	0,5	1	1	5,24	2465,26	2536,69	2517,45
P	15	5	15	0,5	1	3	5,36	2465,91	2521,20	2496,84

C200_P10c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	4,40	2408,47	2503,63	2457,78
A	20	5	15	0,5	1	2	5,31	2441,31	2484,36	2462,63
B	30	5	15	0,5	1	2	7,32	2430,06	2546,88	2478,75
C	15	10	15	0,5	1	2	5,15	2430,14	2498,41	2466,11
D	20	10	15	0,5	1	2	6,33	2417,35	2498,31	2446,87
E	30	15	15	0,5	1	2	8,98	2417,77	2490,26	2456,94
F	15	5	20	0,5	1	2	5,62	2383,09	2456,44	2421,00
G	15	5	30	0,5	1	2	8,22	2361,84	2464,06	2411,49
H	15	5	40	0,5	1	2	10,75	2380,70	2469,84	2416,94
I	20	10	30	0,5	1	2	11,84	2386,21	2449,30	2420,17
J	15	5	15	0,7	1	2	5,17	2425,22	2487,72	2449,61
K	15	5	15	0,9	1	2	4,31	2433,70	2524,57	2464,37
L	15	5	15	0,5	2	2	4,48	2414,34	2549,77	2487,46
M	15	5	15	0,5	3	2	4,45	2459,41	2547,02	2501,99
O	15	5	15	0,5	1	1	4,37	2384,25	2493,99	2438,37
P	15	5	15	0,5	1	3	4,53	2422,59	2504,83	2466,66

Πίνακας 8.8 Αποτελέσματα C 200_P10

Με την αύξηση των πελατών και την αναμενόμενη αύξηση της πολυπλοκότητας ισχύουν οι ίδιες παρατηρήσεις του σεναρίου με 150 πελάτες και 5 θέσεις στάθμευσης. Δηλαδή ο **O** παύει να είναι αποτελεσματικός, και

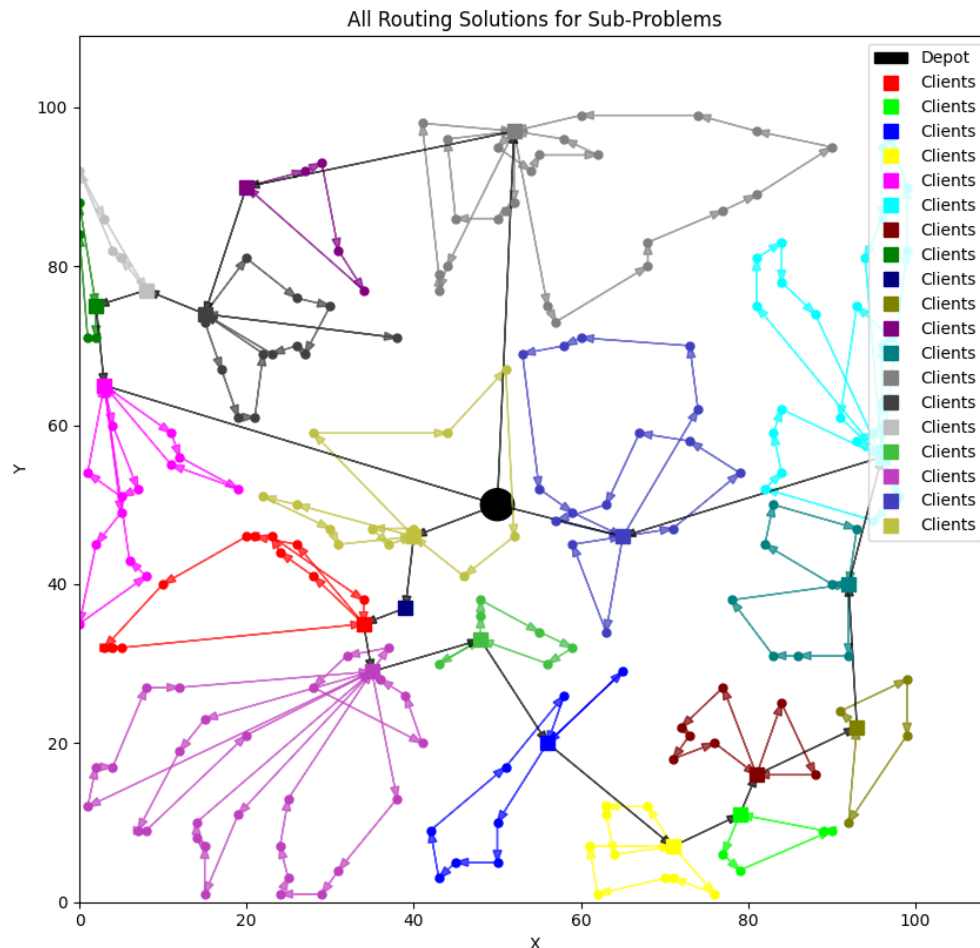
καλές λύσεις παρέχονται μόνο από τους συνδυασμούς με πολλές επαναλήψεις η περισσότερα μυρμήγκια.

8.1.6 Προβλήματα με 200 πελάτες και 20 θέσεις στάθμευσης:

C200_P20a										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	6,52	2346,12	2402,17	2377,64
A	20	5	15	0,5	1	2	7,23	2359,23	2417,24	2386,46
B	30	5	15	0,5	1	2	9,84	2360,66	2417,63	2382,05
C	15	10	15	0,5	1	2	7,31	2348,08	2418,10	2376,85
D	20	10	15	0,5	1	2	8,96	2359,57	2407,42	2381,36
E	30	15	15	0,5	1	2	13,07	2351,00	2381,37	2366,85
F	15	5	20	0,5	1	2	7,81	2344,25	2408,52	2370,37
G	15	5	30	0,5	1	2	10,98	2347,02	2434,38	2368,64
H	15	5	40	0,5	1	2	14,49	2348,78	2395,38	2366,29
I	20	10	30	0,5	1	2	15,93	2349,78	2362,82	2356,06
J	15	5	15	0,7	1	2	7,13	2361,49	2391,35	2379,78
K	15	5	15	0,9	1	2	5,91	2349,74	2403,12	2372,38
L	15	5	15	0,5	2	2	8,15	2370,72	2447,82	2408,80
M	15	5	15	0,5	3	2	6,12	2371,56	2463,28	2426,82
O	15	5	15	0,5	1	1	5,99	2367,06	2400,56	2384,75
P	15	5	15	0,5	1	3	6,14	2352,36	2395,00	2378,24
C200_P20b										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	6,29	2232,31	2277,02	2257,40
A	20	5	15	0,5	1	2	7,27	2255,87	2323,57	2283,26
B	30	5	15	0,5	1	2	9,84	2237,74	2322,39	2279,42
C	15	10	15	0,5	1	2	7,23	2220,65	2274,48	2244,80
D	20	10	15	0,5	1	2	8,70	2235,56	2299,88	2269,24
E	30	15	15	0,5	1	2	12,15	2229,89	2302,98	2258,23
F	15	5	20	0,5	1	2	7,68	2244,97	2317,82	2265,96
G	15	5	30	0,5	1	2	11,02	2237,57	2317,17	2262,29
H	15	5	40	0,5	1	2	14,71	2203,62	2300,94	2249,92
I	20	10	30	0,5	1	2	16,24	2222,42	2279,50	2252,43
J	15	5	15	0,7	1	2	7,23	2225,99	2271,50	2251,78
K	15	5	15	0,9	1	2	6,05	2251,21	2298,74	2273,67
L	15	5	15	0,5	2	2	6,07	2271,17	2334,32	2298,98
M	15	5	15	0,5	3	2	6,17	2266,73	2363,60	2325,20
O	15	5	15	0,5	1	1	6,03	2236,88	2331,67	2282,18
P	15	5	15	0,5	1	3	6,13	2238,89	2308,86	2274,55
C200_P20c										
	Ants	Best ants	Iterations	Decay	Alpha	Beta	Average run time (s)	Best distance (km)	Worst distance (km)	Average distance (km)
Base	15	5	15	0,5	1	2	5,94	2226,75	2258,61	2241,51
A	20	5	15	0,5	1	2	7,13	2195,06	2303,35	2252,42
B	30	5	15	0,5	1	2	9,85	2228,97	2274,75	2249,27
C	15	10	15	0,5	1	2	7,55	2198,03	2261,47	2234,79
D	20	10	15	0,5	1	2	8,61	2201,44	2261,75	2233,09
E	30	15	15	0,5	1	2	11,96	2189,07	2243,00	2221,90
F	15	5	20	0,5	1	2	7,54	2196,81	2278,43	2239,97
G	15	5	30	0,5	1	2	10,92	2217,66	2261,02	2240,99
H	15	5	40	0,5	1	2	14,14	2183,00	2267,31	2232,24
I	20	10	30	0,5	1	2	15,66	2188,69	2255,49	2219,50
J	15	5	15	0,7	1	2	7,06	2178,92	2264,00	2219,02
K	15	5	15	0,9	1	2	5,82	2198,85	2271,98	2232,08
L	15	5	15	0,5	2	2	5,85	2234,94	2299,89	2257,47
M	15	5	15	0,5	3	2	6,19	2237,66	2304,19	2276,34
O	15	5	15	0,5	1	1	6,04	2186,80	2277,22	2214,16
P	15	5	15	0,5	1	3	6,00	2237,51	2296,10	2263,15

Πίνακας 8.9 Αποτελέσματα C 200_P20

Τέλος με την αύξηση των θέσεων στάθμευσης η απόδοση των συνδυασμών που δίνουν λιγότερη έμφαση στην μνήμη του αλγορίθμου αυξάνεται, και πλέον είναι ακόμα πιο φανερό ότι σε μεγάλα προβλήματα γίνεται όλο και πιο συμφέρουσα η τοποθέτηση περισσότερων θέσεων για τα φορτηγά. Πιθανότατα αυτό συμβαίνει επειδή περιορίζεται και η πολυπλοκότητα. Παρακάτω ακολουθεί ένα γράφημα με μια λύση του παραπάνω προβλήματος:



Εικόνα 8.1 Γραφική αναπαράσταση λύσης για C150_P15

Συμπερασματικά διαπιστώνεται ότι η απόδοση του αλγορίθμου βελτιώνεται σημαντικά με την αύξηση του decay στο επίπεδο 0,7, χωρίς να αυξάνεται σημαντικά ο χρόνος επίλυσης. Ο συνδυασμός αυτής της παραμέτρου με περισσότερα μυρμήγκια και επαναλήψεις αναμένεται να προσφέρει ακόμα μικρότερα κόστη. Ακόμα παρατηρείται ότι η αύξηση των θέσεων στάθμευσης λειτουργεί επικουρικά στην απόδοση του αλγορίθμου, καθώς οδηγεί στην περαιτέρω διάσπαση του συνολικού προβλήματος σε περισσότερα μικρότερα άρα και πιο εύκολα επιλύσιμα.

8.2 Δοκιμή διάφορων τύπων μη επανδρωμένων εναέριων οχημάτων

Σε αυτό το υποκεφάλαιο των αποτελεσμάτων, θα εξεταστούν τρία διαφορετικά μοντέλα μη επανδρωμένων εναέριων οχημάτων, καθένα με ξεχωριστά χαρακτηριστικά όσον αφορά την εμβέλεια, την κατανάλωση ενέργειας, την ταχύτητα και τη χωρητικότητα. Σκοπός είναι να αξιολογηθεί η απόδοσή τους σε ένα ρεαλιστικό σενάριο παραδόσεων, ώστε να αναδειχθούν οι επιπτώσεις αυτών των χαρακτηριστικών στην αποτελεσματικότητά τους.





Παράλληλα, επιδιώκεται να εξεταστεί συνολικά η αποδοτικότητα του συστήματος συνεργατικού στόλου σε σχέση με ένα συμβατικό σύστημα παράδοσης. Η ανάλυση αυτή θα βοηθήσει στην εξαγωγή συμπερασμάτων σχετικά με την ενεργειακή αποδοτικότητα και τους χρόνους των δύο προσεγγίσεων, προσφέροντας μια πρώτη εικόνα για τα πλεονεκτήματα και τις αδυναμίες του κάθε συστήματος.

Αξίζει να αναφερθεί ότι το παρακάτω πείραμα έγινε με τις εξής επιπλέον παραδοχές:

- Ο αλγόριθμος έχει την δυνατότητα να ελαχιστοποιεί μόνο την απόσταση. Παρόλα αυτά ο κώδικας που υλοποιήθηκε μπορεί να υπολογίσει επιπρόσθετα τον χρόνο ολοκλήρωσης των δρομολογίων και τη συνολική κατανάλωση ενέργειας
- Ο αλγόριθμος δεν λαμβάνει υπόψιν την διαφορά στην κατανάλωση ενέργειας μεταξύ επίγειων και εναέριων οχημάτων. (Ο αλγόριθμος δεν αναθέτει τους πελάτες στις θέσεις στάθμευσης δυναμικά. Δεν θα μπορούσε ευκολά τροποποιηθεί στο να ελαχιστοποιήσει αποτελεσματικά ως προς την ενέργεια)
- Θεωρείται όμως ότι η ελαχιστοποίηση της απόστασης συνδέεται σε σημαντικό βαθμό με την ελαχιστοποίηση του χρόνου και της καταναλισκόμενης ενέργειας.
- Θεωρείται ότι το φορτίο δεν επηρεάζει την κατανάλωση ενέργειας, και ως κατανάλωση θεωρούμε την κατανάλωση ανά χιλιόμετρο του οχήματος με μέγιστο φορτίο
- Δεν συμπεριλαμβάνονται στους υπολογισμούς οι χρόνοι απογείωσης, προσγείωσης, και παράδοσης φορτίου
- Στα συγκεκριμένα προβλήματα οι θέσεις στάθμευσης δεν ορίζονται τυχαία αλλά σαν τα κεντροειδή που προκύπτουν από συσταδοποίηση

k -means με μια μικρή τυχαία μετακίνηση. Η συγκεκριμένη επιλογή έγινε ώστε τα αποτελέσματα να μην επηρεάζονται τόσο πολύ από την τυχαιότητα των θέσεων στάθμευσης , μιας και αναγνωρίζεται η αδυναμία του αλγορίθμου για βέλτιστη ανάθεση. Ακόμα η επιλογή αυτή προσομοιώνει καλύτερα ένα ρεαλιστικό σενάριο, στο οποίο η τοποθέτηση τους θα είχε γίνει στρατηγικά.

8.2.1 Χαρακτηριστικά οχημάτων

Drone	Image	Payload (kg)	Consumption kWh/km	Range km	Speed km/h
Wingcopter 198		5	2,2	75	100
Dji FlyCart Single Battery		30	24,8	16	50
Dji FlyCart Double Battery		40	24,8	8	50
FreeFly Alta X		9	6	24	65

Πίνακας 8.10 Χαρακτηριστικά εναέριων οχημάτων

Τα τρία αυτά μοντέλα αντιπροσωπεύουν διαφορετικές κατηγορίες μη επανδρωμένων εναέριων οχημάτων (UAV), καθεμία με τα δικά της χαρακτηριστικά και πλεονεκτήματα.

Το Wingcopter 198 είναι ένα UAV τύπου VTOL (Vertical Take-Off and Landing). Αυτά τα οχήματα συνδυάζουν την ικανότητα αιώρησης, όπως τα κλασικά drones, με τη δυνατότητα πτήσης σαν αεροσκάφη σταθερής πτέρυγας . Αυτή η υβριδική λειτουργία τους επιτρέπει να προσγειώνονται και να απογειώνονται εύκολα, ενώ ταυτόχρονα διαθέτουν μεγάλη εμβέλεια.

Το DJI FlyCart είναι ένα οκτακόπτερο (δηλαδή έχει οκτώ έλικες, δύο σε κάθε βραχίονα, πάνω και κάτω) που του προσδίδουν τη δυνατότητα μεταφοράς πολύ μεγάλων φορτίων. Αυτή η αυξημένη ικανότητα ανύψωσης επιτυγχάνεται εις βάρος της εμβέλειάς του, καθιστώντας το πιο κατάλληλο


για μικρότερες αποστάσεις. Το FlyCart μπορεί να λειτουργήσει είτε με μία μπαταρία, για να μεγιστοποιηθεί το βάρος που μπορεί να μεταφέρει, είτε με δύο μπαταρίες, για να αυξηθεί η εμβέλεια του.

Το Alta X, το τρίτο μοντέλο, είναι ένα τετρακόπτερο (με τέσσερις έλικες) , δηλαδή το πιο δημοφιλές είδος drone λόγω του χαμηλού κόστους, της απλότητας και της αξιοπιστίας του. Το συγκεκριμένο μοντέλο διαθέτει ικανοποιητική εμβέλεια και είναι σε θέση να μεταφέρει σχετικά μεγάλα φορτία.

Τα παραπάνω χαρακτηριστικά και οι εικόνες συλλέχθηκαν κυρίως από τις ιστοσελίδες των κατασκευαστών (<https://freeflysystems.com>, 2025), (<https://motionair.no>, 2019), (<https://www.dji.com>, 2023), (<https://www.synerjet.com.br>, 2021). Ωστόσο όσον αφορά την κατανάλωση ενέργειας ανά χιλιόμετρο έχουν γίνει κάποιες εκτιμήσεις καθώς οι συγκεκριμένες πληροφορίες δεν ήταν διαθέσιμες. Παρόλα αυτά ήταν γνωστές οι χωρητικότητες των μπαταριών και η εμβέλεια υπό φορτίο οπότε η κατανάλωση υπολογίστηκε ως εξής:

$$\text{Κατανάλωση Ενέργειας (Kwh/km)} = \frac{\text{Χωρητικότητα_Μπαταρίας (Kwh)}}{\text{Εμβέλεια (km)}}$$

Για την μελέτη αυτή σαν επίγειο όχημα θα χρησιμοποιηθεί το VW ABT e-Transporter, τα χαρακτηριστικά του είναι :

Φορτηγό	Εικόνα	Payload kg	Consumption kWh/km	Range km	Speed km/h
VW ABT e-Transporter		855	35.6	122	50

Πίνακας 8.11 Χαρακτηριστικά επίγειου οχήματος

Η εικόνες και τα δεδομένα προέρχονται από την ιστοσελίδα του κατασκευαστή: (<https://www.volkswagenfans>, 2025). Είναι σημαντικό να διευκρινιστεί ότι η μέγιστη ταχύτητα του είναι φυσικά πολύ μεγαλύτερη. Στο πλαίσιο της εν λόγω μελέτης όμως θεωρείται πως η μέση ταχύτητα με την οποία θα μπορεί να κινείται στους δρόμους είναι 50 χιλιόμετρα την ώρα.

8.2.2 Μεθοδολογία δημιουργίας προβλημάτων

Για τις ανάγκες του συγκεκριμένου πειράματος και προκειμένου να προσδιοριστούν τα επιθυμητά χαρακτηριστικά για τη χρήση drones, καθώς και να δοκιμαστεί η ιδέα του συνεργατικού στόλου φορτηγών και drones, δημιουργήθηκε μια σειρά από διαφορετικά σενάρια. Τα σενάρια αυτά περιλάμβαναν ποικιλίες στον αριθμό των drones, των πελατών και των θέσεων στάθμευσης, ώστε να αξιολογηθεί η αποδοτικότητα της προτεινόμενης μεθόδου.

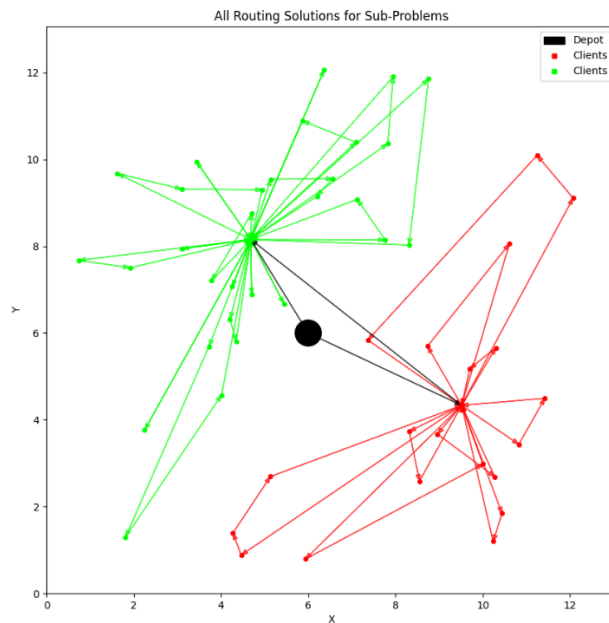
Όλα τα προβλήματα εκτυλίσσονται μέσα σε έναν χώρο 13 x 13 χιλιομέτρων, που αντιστοιχεί στο μέγεθος μιας μικρής πόλης, όπως τα Χανιά, για να προσομοιωθεί ένα ρεαλιστικό περιβάλλον. Επιπλέον, η ζήτηση των πελατών κυμάνθηκε από ένα έως τρία κιλά, προσομοιώνοντας παραδόσεις μικρού μεγέθους, που είναι ιδανικές για τη χρήση drone.

8.2.3 Σενάρια με 50 πελάτες

Clients	P_Spots	Drone model	Fleet size	Distance (km)	Consumption (kW)	Time (h)	Feasible solution
50	2	Wingcopter 198	2	196,59	855,17	0,81	Ναι
50	2	Wingcopter 198	3	195,71	853,24	0,62	Ναι
50	2	Wingcopter 198	4	195,35	852,45	0,51	Ναι
50	2	FlyCart 30	2	93,96	2468,57	1,14	Ναι
50	2	FlyCart 30	3	93,96	2468,57	0,97	Ναι
50	2	FlyCart 30	4	93,05	2446,08	0,95	Ναι
50	2	FlyCart 30 Single Battery	2	73,66	1965,08	0,80	Όχι
50	2	FlyCart 30 Single Battery	3	77,18	2052,54	0,63	Όχι
50	2	FlyCart 30 Single Battery	4	79,01	2097,81	0,59	Όχι
50	2	Alta X	2	129,06	1149,25	0,93	Ναι
50	2	Alta X	3	128,56	1146,23	0,74	Ναι
50	2	Alta X	4	127,88	1142,17	0,64	Ναι
50	4	Wingcopter 198	2	149,98	1117,73	0,89	Ναι
50	4	Wingcopter 198	3	151,09	1120,17	0,75	Ναι
50	4	Wingcopter 198	4	150,27	1118,36	0,71	Ναι
50	4	FlyCart 30	2	94,44	2599,96	1,49	Ναι
50	4	FlyCart 30	3	97,08	2665,50	1,50	Ναι
50	4	FlyCart 30	4	96,04	2639,67	1,56	Ναι
50	4	FlyCart 30 Single Battery	2	102,02	2788,05	1,14	Ναι
50	4	FlyCart 30 Single Battery	3	102,02	2788,05	1,12	Ναι
50	4	FlyCart 30 Single Battery	4	102,02	2788,05	1,12	Ναι
50	4	Alta X	2	114,39	1385,01	1,03	Ναι
50	4	Alta X	3	114,52	1385,80	0,91	Ναι
50	4	Alta X	4	114,28	1384,38	0,85	Ναι

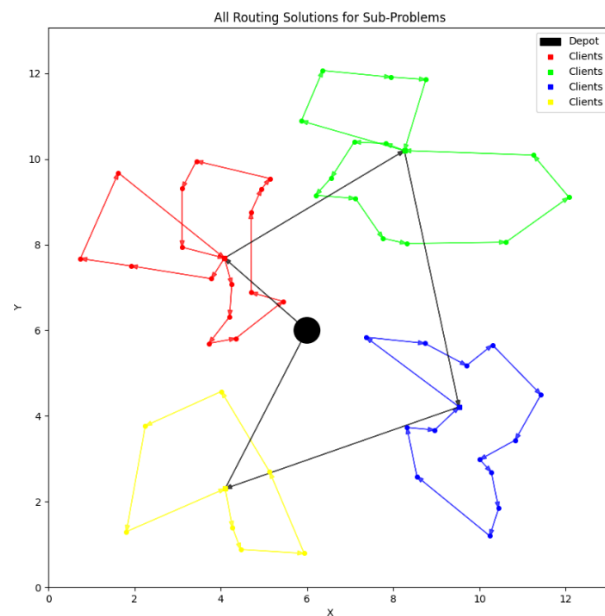
Πίνακας 8.12 Σενάρια με 50 πελάτες

Από τα αποτελέσματα προκύπτει ξεκάθαρα ότι το Wingcopter 198 αποτελεί την καλύτερη επιλογή σε όλα τα κριτήρια, παρά τη σχετικά μικρή χωρητικότητά του. Ωστόσο, παρατηρούνται σημαντικές διαφορές στη συνολική απόσταση που διανύεται, λόγω της ανάγκης συχνών επιστροφών στη βάση για ανεφοδιασμό όπως φαίνεται στην Εικόνα 8.2. Παρόλα αυτά, η εξαιρετικά υψηλή ταχύτητά του, σε συνδυασμό με τη χαμηλή κατανάλωση ενέργειας, το καθιστούν τόσο αποδοτικό που υπερτερεί έναντι των άλλων επιλογών.



Εικόνα 8.2 Wingcopter 198 με 50 πελάτες και 2 θέσεις στάθμευσης

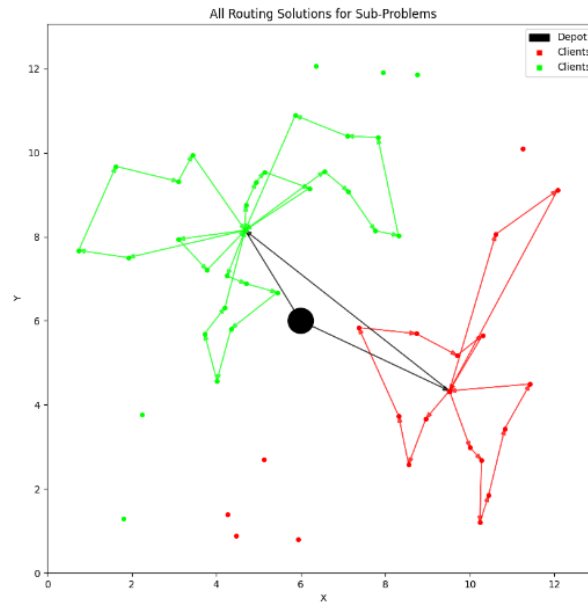
Τις μικρότερες συνολικές διαδρομές διανύει το FlyCart30, όπως αποτυπώνεται στο παρακάτω διάγραμμα, γεγονός που αποδίδεται στη μεγάλη χωρητικότητά του. Η αυξημένη δυνατότητα μεταφοράς φορτίου μειώνει την ανάγκη για συχνές επιστροφές στη βάση.



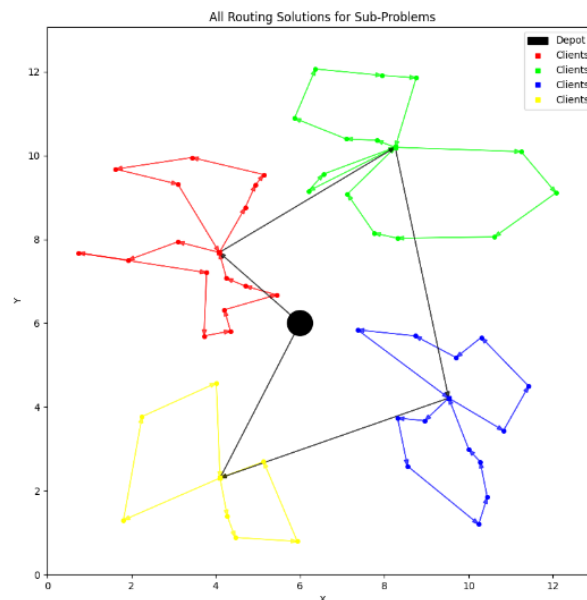
Εικόνα 8.3 Fly Cart 30 Double Battery με 50 πελάτες και 4 θέσεις στάθμευσης

Παρατηρείται, επίσης, ότι η αύξηση των θέσεων στάθμευσης επιδεινώνει τα αποτελέσματα, ιδίως όσον αφορά τον χρόνο και την κατανάλωση, καθώς καλύπτονται μεγαλύτερες αποστάσεις από το φορτηγό, το οποίο κινείται με

χαμηλότερη ταχύτητα και καταναλώνει περισσότερη ενέργεια. Εξαίρεση αποτελεί το FlyCart30 όταν λειτουργεί με μία μπαταρία, καθώς η παρουσία περισσότερων θέσεων στάθμευσης διευκολύνει την πρόσβασή του σε όλους τους πελάτες (Εικόνα 8.5). Αντίθετα με δύο θέσεις στάθμευσης, οι περισσότεροι πελάτες βρίσκονται εκτός της εμβέλειάς του (Εικόνα 8.4).



Εικόνα 8.4 Fly Cart 30 Single Battery με 50 πελάτες και 2 θέσεις στάθμευσης



Εικόνα 8.5 Fly Cart 30 Single Battery με 50 πελάτες και 4 θέσεις στάθμευσης

Το συγκεκριμένο φαινόμενο παρουσιάζει ιδιαίτερο ενδιαφέρον, καθώς φαίνεται ότι ο περιορισμένος αριθμός θέσεων στάθμευσης συμβάλλει στη μείωση του συνολικού κόστους. Ωστόσο, όταν ο αριθμός των θέσεων στάθμευσης μειώνεται κάτω από ένα κρίσιμο όριο, προκύπτει το ζήτημα της

απόστασης, καθώς κάποιοι πελάτες βρίσκονται εκτός εμβέλειας των εναέριων οχημάτων.

Επιπλέον, παρατηρείται ότι η αύξηση του μεγέθους του στόλου βελτιώνει σημαντικά τον συνολικό χρόνο παράδοσης, ειδικά για τα drones που εκτελούν πολλαπλές διαδρομές από κάθε θέση στάθμευσης. Αντίθετα, για το FlyCart30, η αύξηση του στόλου δεν φαίνεται να έχει ουσιαστική επίδραση, τουλάχιστον για σενάρια που περιλαμβάνουν έως 50 πελάτες

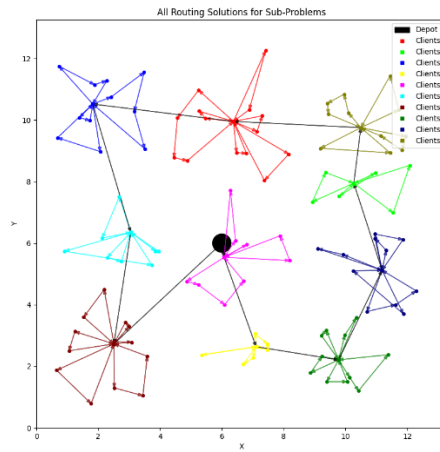
8.2.4 Σενάρια με 100 πελάτες

Τα αποτελέσματα με 100 πελάτες φαίνονται παρακάτω:

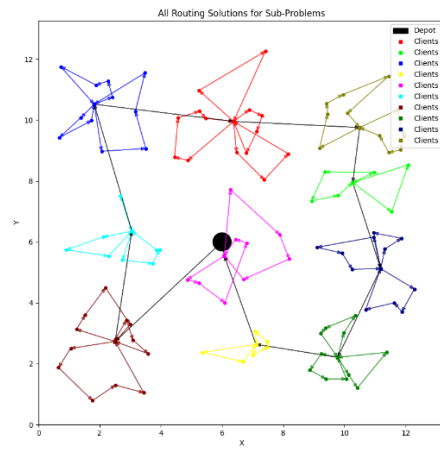
Clients	P_Spots	Drone model	Fleet size	Distance (km)	Consumption (kW)	Time (h)	Feasible solution
100	5	Wingcopter 198	2	240,9040305	1497,54272	1,272018635	Ναι
100	5	Wingcopter 198	3	237,9917392	1491,135679	1,023704091	Ναι
100	5	Wingcopter 198	4	238,980859	1493,311743	0,937077252	Ναι
100	5	FlyCart 30	2	137,641212	3730,260759	1,846272227	Ναι
100	5	FlyCart 30	3	137,4823286	3726,320451	1,869890907	Ναι
100	5	FlyCart 30	4	136,917812	3712,320441	1,867888479	Ναι
100	5	FlyCart 30 Single Battery	2	147,0604909	3963,858877	1,811953265	Ναι
100	5	FlyCart 30 Single Battery	3	146,7727966	3956,724058	1,4281365	Ναι
100	5	FlyCart 30 Single Battery	4	148,112838	3989,957085	1,435172234	Ναι
100	5	Alta X	2	179,7445885	1936,595651	1,483296665	Ναι
100	5	Alta X	3	180,1344519	1938,934831	1,222417697	Ναι
100	5	Alta X	4	180,7393645	1942,564307	1,129184467	Ναι
100	10	Wingcopter 198	2	193,0647077	1627,310886	1,288603895	Ναι
100	10	Wingcopter 198	3	194,2385679	1629,893378	1,141710126	Ναι
100	10	Wingcopter 198	4	195,7193722	1633,151147	1,093643204	Ναι
100	10	FlyCart 30	2	137,6001935	3806,182829	2,599535762	Ναι
100	10	FlyCart 30	3	137,6001935	3806,182829	2,60862458	Ναι
100	10	FlyCart 30	4	137,6001935	3806,182829	2,582370714	Ναι
100	10	FlyCart 30 Single Battery	2	145,837372	4010,464855	2,0416105	Ναι
100	10	FlyCart 30 Single Battery	3	146,2823501	4021,500311	2,052343481	Ναι
100	10	FlyCart 30 Single Battery	4	145,9878019	4014,195518	2,054898963	Ναι
100	10	Alta X	2	166,7120065	2066,835794	1,608798411	Ναι
100	10	Alta X	3	167,6642206	2078,072427	1,492928646	Ναι
100	10	Alta X	4	166,2904896	2064,306692	1,437554655	Ναι

Πίνακας 8.13 Σενάρια με 100 πελάτες

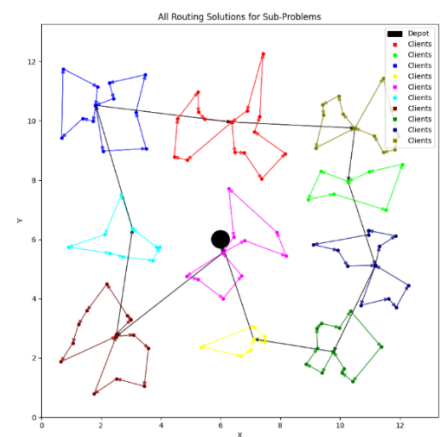
τα προβλήματα με 100 πελάτες, δεν παρατηρούνται διαφοροποιήσεις στα μοτίβα που αναφέρθηκαν στα προηγούμενα αποτελέσματα. Η μόνη αξιοσημείωτη διαφορά είναι ο σχεδόν διπλασιασμός του συνολικού κόστους. Στη συνέχεια, παρουσιάζονται ενδεικτικά ορισμένες λύσεις:



Εικόνα 8.6 Wingcopter 198 με 100 πελάτες και 10 θέσεις στάθμευσης



Εικόνα 8.7 Alta X με 100 πελάτες και 10 θέσεις στάθμευσης



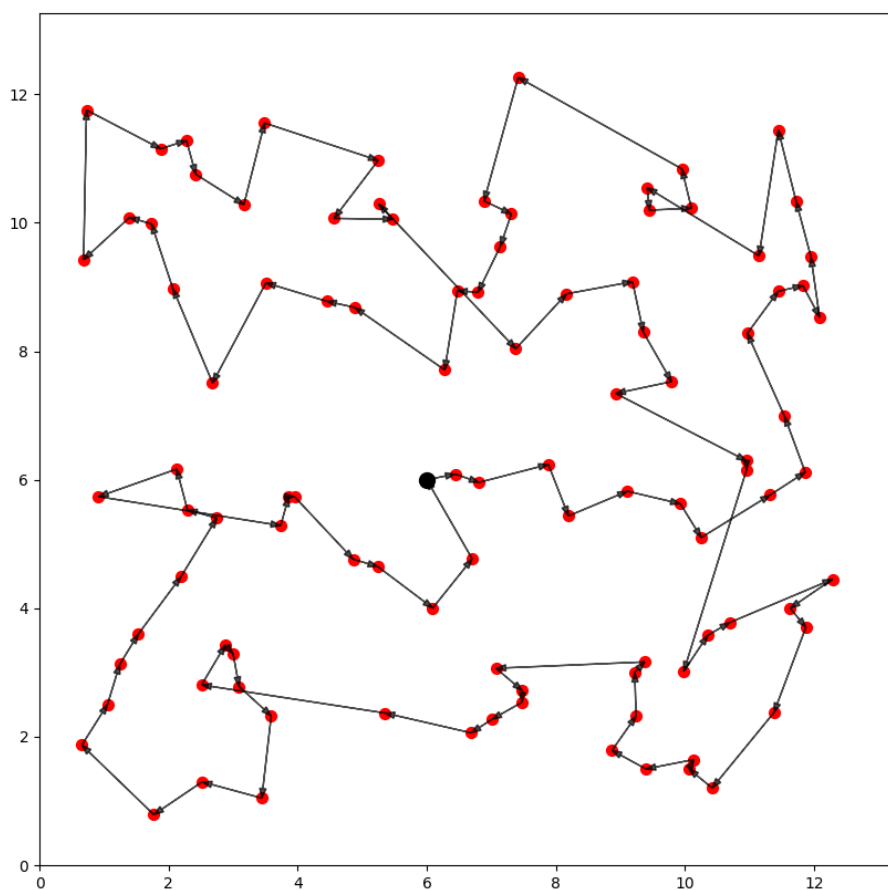
Εικόνα 8.8 FlyCart30 Double Battery με 100 πελάτες και 10 θέσεις στάθμευσης

8.2.5 Σύγκριση με συμβατικά σενάρια

Φυσικά, για την εξαγωγή αξιόπιστων συμπερασμάτων σχετικά με το σύστημα συνεργατικού στόλου, τα αποτελέσματα πρέπει να συγκριθούν με εκείνα ενός συμβατικού συστήματος παράδοσης που βασίζεται αποκλειστικά σε φορτηγά. Στην παρακάτω ενότητα παρουσιάζονται τα αποτελέσματα για 50 και 100 πελάτες, συνοδευόμενα από μια ενδεικτική γραφική αναπαράσταση λύσεων:

Clients	Fleet size	Distance (km)	Consumption (kW)	Time (h)
100	1	137,60	3806,18	2,60
50	1	93,01	2444,99	1,08

Πίνακας 8.14 Αποτελέσματα συμβατικών σεναρίων για 50 και 100 πελάτες



Εικόνα 8.9 Διαδρομή φορτηγού για 100 πελάτες

Συγκρίνοντας τα αποτελέσματα, παρατηρείται ότι η συμβατική λύση είναι σχεδόν εξίσου αποδοτική με το χειρότερο σενάριο συνεργασίας εναέριου οχήματος. Παράλληλα, τα καλύτερα σενάρια για 50 και 100 πελάτες αντίστοιχα είναι ιδιαίτερα ανταγωνιστικά ως προς τον χρόνο και την κατανάλωση ενέργειας. Παρακάτω παρουσιάζονται οι συγκρίσεις:

System	Clients	P_Spots	Drone model	Fleet size	Distance (km)	Consumption (kW)	Time (h)
Truck only	50	None	None	None	93,96	2444,99	1,14
Best with drones	50	2	Wingcopter 198	4	195,35	852,45	0,51
Worst with drones	50	4	Cart 30 Single Batt	2	102,02	2788,05	1,14
Truck only	100	None	None	None	196,59	3806,18	0,81
Best with drones	100	5	Wingcopter 198	4	238,98	1493,31	0,94
Worst with drones	100	10	FlyCart 30	2	137,60	3806,18	2,60

Πίνακας 8.15 Σύγκριση συμβατικού και συνεργατικού συστήματος

Ωστόσο, είναι σημαντικό να σημειωθεί ότι αν γινόταν χρήση περισσότερων φορητών, πιθανότατα οι παραδόσεις θα ολοκληρώνονταν σε εξίσου καλούς χρόνους. Ακόμα είναι φανερό ότι ο αλγόριθμος 'δυσκολεύεται' να επιλύσει το πρόβλημα, καθώς στην εικόνα 8.9 παρατηρούνται διασταυρώσεις των διαδρομών που σε γενικές γραμμές είναι ενδεικτικό «κακών» λύσεων. Το συμπέρασμα που προκύπτει είναι ότι το σύστημα συνεργατικού στόλου μπορεί να είναι ανταγωνιστικό σε σχέση με τα συμβατικά μέσα κυρίως όσον αφορά την κατανάλωση ενέργειας, τουλάχιστον για τα δεδομένα και τις παραδοχές της συγκεκριμένης προσομοίωσης.

Είναι σημαντικό επίσης να τονιστεί ότι οι χρόνοι που παρουσιάζονται δεν είναι ρεαλιστικοί, καθώς προφανώς θα ήταν αδύνατον να εξυπηρετηθούν 100 πελάτες σε λιγότερο από μια ώρα. Ωστόσο οι παραδοχές για την προσομοίωση των δυο συστημάτων είναι κοινές, και ως εκ τούτου οι συγκρίσεις που γίνονται ανταποκρίνονται στην πραγματικότητα.

Συμπεράσματα

Στην παρούσα διπλωματική παρουσιάστηκε το πρόβλημα δρομολόγησης συνεργατικού στόλου επίγειων και μη επανδρωμένων εναέριων οχημάτων και η επίλυση του με τον αλγόριθμο αποικίας μυρμηγκιών (ACO). Παρόλο που η μέθοδος επίλυσης που προτείνεται είναι αρκετά απλή, και περιλαμβάνει τη διάσπαση του προβλήματος σε πολλά μικρότερα, τα αποτελέσματα είναι ικανοποιητικά ειδικά αν οι θέσεις στάθμευσης έχουν τοποθετηθεί με στρατηγικό τρόπο. Επιπρόσθετα ενώ η συγκεκριμένη υλοποίηση δεν μπορεί να βρει την βέλτιστη λύση, είναι εύκολο να τροποποιηθεί πάνω σε νέα δεδομένα και λειτουργίες. Για παράδειγμα ο κεντρικός αλγόριθμος επίλυσης ACO μπορεί εύκολα να αντικατασταθεί από οποιονδήποτε άλλο αλγόριθμο μπορεί να λύσει το απλό πρόβλημα της δρομολόγησης οχημάτων με περιορισμό χωρητικότητας και χρόνου.

Επιπλέον παρατηρήθηκε ότι ο μεγαλύτερος και σημαντικότερος περιορισμός των εναέριων οχημάτων είναι οι εμβέλεια τους, καθώς χωρίς την υποστήριξη από ενδιάμεσους σταθμούς είναι δύσκολο να φτάσουν σε όλους τους πελάτες. Μάλιστα αυτό το πρόβλημα αναδύθηκε και στο σενάριο μιας μικρής πόλης όπως τα Χανιά και φυσικά είναι αναμενόμενο να γίνει πολύ πιο έντονο σε μεγαλύτερα αστικά κέντρα. Για αυτό και θα ήταν ωφέλιμο, οι κατασκευαστές μη επανδρωμένων να εστιάσουν σε οχήματα με μεγάλη εμβέλεια και ταχύτητα, όπως τα οχήματα τύπου VTOL παρά σε οχήματα που σηκώνουν βαριά φορτία όπως τα οκτακόπτερα.

Τέλος η σύγκριση των συμβατικών μοντέλων παραδόσεων με το προτεινόμενο προσέφερε σημαντικές πληροφορίες σχετικά με τον σχεδιασμό συστημάτων διανομής που ενσωματώνουν μη επανδρωμένα εναέρια οχήματα. Πιο συγκεκριμένα διαπιστώθηκε ότι είναι πολύ πιθανό ένα συμβατικό σύστημα να αποδειχτεί πιο αποδοτικό, αν δεν γίνει η σωστή επιλογή των προδιαγραφών των drone. Μάλιστα ένα κομβικό ερώτημα είναι αν οι βελτιωμένοι χρόνοι παράδοσης και η ελαχιστοποίηση της καταναλισκόμενης ενέργειας που υπόσχονται αυτές οι καινοτόμες λύσεις μπορούν να αντισταθμίσουν το υψηλό κόστος και ρίσκο της υλοποίησης τους.

Τα παραπάνω συμπεράσματα όμως θα πρέπει να πλαισιωθούν στις υποθέσεις που θεωρήθηκαν κατά την εκτέλεση της παρούσας εργασίας. Για την μοντελοποίηση του προβλήματος θεωρήθηκαν πολλές παραδοχές, που το καθιστούν στην παρούσα φάση της τεχνολογίας, μη ρεαλιστικό. Για παράδειγμα θεωρείται ότι η επαναφόρτιση των οχημάτων ή η αλλαγή μπαταρίας γίνεται σε αμελητέο χρόνο και ακόμα ότι τα εναέρια οχήματα

έχουν θήκες που μπορούν να αποθηκεύσουν πακέτα ανεξάρτητα με τον αριθμό και τον όγκο τους.

Σε μελλοντικές έρευνες πάνω στους συνεργατικούς στόλους διανομής θα μπορούσαν να εξεταστούν αυτοί οι περιορισμοί σε μοντέλα όπου τα επίγεια οχήματα θα είχαν την δυνατότητα να μεταφέρουν πολλαπλές μπαταρίες κάποιες από τις οποίες θα μπορούν να φορτώνουν στα εναέρια και κάποιες από τις οποίες θα μπορούσαν να φορτίζουν για χρήση αργότερα. Ακόμα θα μπορούσαν να εξεταστούν ανάμεσα σε άλλα, η δυνατότητα προσγείωσης μη επανδρωμένου σε άλλο φορτηγό, οι περιορισμοί καιρικών συνθηκών και οι μεταβολές στην κατανάλωση με βάση την ταχύτητα, το υψόμετρο και το φορτίο.

Βιβλιογραφία

- Anne Goodchild, J. T. (2018, Ιούνιος). Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO2 emissions in the delivery service industry. *Transportation Research Part D: Transport and Environment, Volume 61, Part A*, σσ. 58-67.
- Emrah Demir, A. S. (2022, Οκτώβριος). Last mile logistics: Research trends and needs. *IMA Journal of Management Mathematics, Volume 33, Issue 4*, σσ. 549-561.
- Giusy Macrina, L. D. (2020, Νοέμβριος). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies, Volume 120*.
<https://freeflysystems.com>. (2025). Ανάκτηση από <https://freeflysystems.com/alta-x/specs>.
- <https://motionair.no>. (2019). Ανάκτηση από <https://motionair.no/drones/>.
- <https://www.dji.com>. (2023). Retrieved from <https://www.dji.com/gr/flycart-30/specs>.
- <https://www.synerjet.com.br>. (2021). Ανάκτηση από <https://www.synerjet.com.br/wingcoptertudoeng.html>.
- <https://www.volkswagenvans>. (2025). Ανάκτηση από <https://www.volkswagenvans.ie/en.html>.
- Kitjacharoenchai, P. M.-C. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *Int. J. Prod. Econ.* 225.
- Marco Dorigo, T. S. (2004). *Ant Colony Optimization*. Bradford Books.
- Michel Gendreau, J.-Y. P. (2003). *Handbook of Metaheuristics 2nd ed*. Springer.
- Nikolaos A. Kyriakakis, T. S. (2022). The electric vehicle routing problem with drones: An energy minimization approach for aerial deliveries. *Cleaner Logistics and Supply Chain Volume 4*.
- O. Deepa, D. A. (2016, Μάρτιος). SWARM INTELLIGENCE FROM NATURAL TO ARTIFICIAL SYSTEMS: ANT COLONY OPTIMAZATION. *International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks (GRAPH-HOC) Vol.8, No.1*.

- Pina-Pardo, J. S. (2021). The traveling salesman problem with release dates and drone resupply. *Comput. Oper. Res.* 129.
- Raj, R. M. (2020). The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transp. Res. Part C* 120.
- Shahzaad, B. B. (2021). Resilient composition of drone services for delivery. *Fut. Gener. Comput. Syst.* 115, σσ. 335-350.
- Sunil Chopra, P. M. (2007). *Supply Chain Management*, 3rd ed. Upper Saddle River, New Jersey: Pearson Prentice Hall.
- T. Tavares, V. B. (2019). *Unmanned Airspace*. Ανάκτηση από <https://www.unmannedairspace.info/commentary/comparing-the-cost-effectiveness-of-drones-v-ground-vehicles-for-medical-food-and-parcel-deliveries/>.
- U. Breunig, V. S. (2016, Δεκέμβριος). A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research volume* 76, σσ. 208-225.
- Vipul Garg a, S. N. (2023, Οκτώβριος). Drones in last-mile delivery: A systematic review on Efficiency, Accessibility, and Sustainability. *Transportation Research Part D: Transport and Environment*.
- Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. *Stochastic Algorithms: Foundations and Applications, SAGA 2009*,.
- Zhen, L. L. (2019). A vehicle routing problem arising inunmanned aerial monitoring. *Comput. Oper. Res.* 105, 1–11.
- Ιωάννης Μαρινάκης, Α. Μ. (2008). *Σχεδιασμός και Βελτιστοποίηση Εφοδιαστικής Αλυσίδας*. Θεσσαλονίκη: Εκδόσεις Σοφία.