

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ
ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΑΥΤΟΝΟΜΑ ΕΝΑΕΡΙΑ
ΟΧΗΜΑΤΑ (DRONES)**

Διπλωματική Εργασία

του

Κωνσταντίνος Καραγιώργος

Θεσσαλονίκη , 24/02/2021

ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ
ΟΧΗΜΑΤΩΝ ΜΕ ΑΥΤΟΝΟΜΑ ΕΝΑΕΡΙΑ ΟΧΗΜΑΤΑ (DRONES)

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής

Άγγελος Σιφαλέρας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24/02/2021

Άγγελος Σιφαλέρας

Σαμαράς Νικόλαος

Χρήστου - Βαρσακέλης
Δημήτριος

.....

Κωνσταντίνος Καραγιώργος

.....

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως σκοπό την παρουσίαση της υλοποίησης και εφαρμογής μεθευρετικής μεθόδου για την εύρεση λύσης σε πρόβλημα βελτιστοποίησης. Ειδικότερα χρησιμοποιείται η μεθευρετική μέθοδος αναζήτηση μεταβλητής γειτνίασης σε ένα πρόβλημα βελτιστοποίησης που ανήκει στην κατηγορία δρομολόγησης οχημάτων όπου σαν όχημα χρησιμοποιείται αυτόνομο εναέριο όχημα. Για την προγραμματιστική υλοποίηση του προβλήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού python. Αρχικά γίνεται μία παρουσίαση της βιβλιογραφίας του προβλήματος και των αλγορίθμων που έχουν χρησιμοποιηθεί για την επίλυση το, επίσης γίνεται μία αναφορά στα αυτόνομα εναέρια οχήματα, τις κατηγορίες και τους τεχνικούς περιορισμούς τους. Στην συνέχεια παρουσιάζεται η προτεινόμενη μεθοδολογία επίλυσης για το πρόβλημα. Τέλος γίνεται μία παρουσίαση των αποτελεσμάτων της εφαρμογής της προτεινόμενης μεθοδολογίας στο πρόβλημα και εξαγωγή συμπερασμάτων.

Λέξεις Κλειδιά: αναζήτηση μεταβλητής γειτνίασης, πρόβλημα δρομολόγησης οχημάτων, αυτόνομο εναέριο όχημα

Abstract

This thesis aims to present the design and implementation of a metaheuristic method for the solution of an optimization problem. The Variable Neighborhood search metaheuristic is used to solve a Vehicle routing problem. Drones were used as vehicle of the problem. To implement the algorithms python programming language was used. The first part of the thesis presents the bibliography of the Travelling Salesman Problem, the Vehicle Routing Problem, the types of commercial drones and a presentation of algorithmic solutions to VRP problem. At the last part of the thesis there is a presentation of the results of the algorithms applied to the problem.

Keywords: vns, vrp, drones

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω από καρδιάς τον επιβλέποντα καθηγητή κ. Άγγελο Σιφαλέρα για την βοήθεια και την υπομονή που έδειξε για την ολοκλήρωση αυτής της εργασίας

Αφιερώνεται στον Πατέρα μου...

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	1
1.1	ΠΡΟΒΛΗΜΑ - ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΟΥ ΘΕΜΑΤΟΣ	1
1.2	ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ	1
1.3	ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΜΕΛΕΤΗΣ	2
1.4	ΣΥΝΕΙΣΦΟΡΑ	2
2	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	3
2.1	ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ	3
2.1.1	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	4
2.1.2	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	5
2.1.3	ΣΥΜΜΕΤΡΙΚΟ TSP	5
2.1.4	ΜΗ ΣΥΜΜΕΤΡΙΚΟ TSP	6
2.1.5	ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ TSP	7
2.2	VRP	7
2.2.1	ΕΙΣΑΓΩΓΗ	7
2.2.2	ΒΑΣΙΚΗ ΜΟΡΦΗ ΠΡΟΒΛΗΜΑΤΟΣ	8
2.2.3	ΠΑΡΑΛΛΑΓΕΣ ΚΛΑΣΣΙΚΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	9
2.3	DRONES	11
2.3.1	ΚΑΤΗΓΟΡΙΕΣ DRONES	11
3	ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ VRP	18
3.1	ΜΕΘΟΔΟΙ ΑΡΧΙΚΟΠΟΙΗΣΗΣ	18
3.2	ΜΕΘΟΔΟΙ ΒΕΛΤΙΩΣΗΣ	19
3.2.1	RELOCATE	19
3.2.2	2-OPT	19
3.3	ΜΕΘΕΥΡΕΤΙΚΕΣ ΜΕΘΟΔΟΙ	20
3.3.1	TABU SEARCH	20
3.3.2	SIMULATED ANNEALING	21
3.3.3	ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	23

3.3.4	VARIABLE NEIGHBORHOOD SEARCH (VNS)	26
4	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ ΛΥΣΗΣ	30
4.1	ΕΙΣΑΓΩΓΗ	30
4.2	ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	30
4.3	ΜΕΘΟΔΟΙ ΑΡΧΙΚΟΠΟΙΗΣΗΣ	32
4.3.1	ΠΡΩΤΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ	32
4.3.2	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	34
4.3.3	ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ	35
4.3.4	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	35
4.3.5	ΤΡΙΤΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ	37
4.3.6	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	37
4.4	ΜΕΘΟΔΟΙ ΒΕΛΤΙΩΣΗΣ	38
4.4.1	ΠΡΩΤΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ	38
4.4.2	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	40
4.4.3	ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ	41
4.4.4	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	42
4.4.5	ΤΡΙΤΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ	43
4.4.6	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	45
5	ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕΛΕΤΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ	47
5.1	ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕΛΕΤΗ	47
5.2	ΑΠΟΤΕΛΕΣΜΑΤΑ	48
5.3	ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND	54
5.3.1	ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ	54
5.3.2	ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ	55
5.3.3	ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ	56
6	ΕΠΙΛΟΓΟΣ	57
6.1	ΣΥΜΠΕΡΑΣΜΑΤΑ	57
6.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	57

Κατάλογος Σχημάτων

2.1.1	TSP	4
2.1.2	SYMETRIC TSP	6
2.1.3	ASYMETRIC TSP	6
2.1.4	TSP MATHEMATICAL MODEL	7
2.2.5	CLASSIC - VEHICLE - ROUTING - PROBLE (VRP)	8
2.2.6	VRP MODEL	9
2.3.7	ΜΗ ΕΠΙΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΑΕΡΟΣΚΑΦΗ	11
2.3.8	FIXED WING	12
2.3.9	FIXED WING DRONE EXAMPLE	13
2.3.10	MIXED WING EXAMPLE DRONE CHARACTERISTICS	13
2.3.11	ROTARY WING DRONE EXAMPLE	14
2.3.12	ROTARY WING DRONE CHARACTERISTICS	15
2.3.13	MULRI-ROTOR	16
2.3.14	D6B - AW - ARM	16
2.3.15	D6B - AW - ARM DRONE CHARACTERISTICS	17
3.2.1	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ RELOCATE INTER - ROUTE ΣΕ VRPTWSD ΠΡΟ- ΒΛΗΜΑ	19
3.2.2	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ 2 - OPT INTRA ROUTE ΣΕ CVRP	20
3.3.3	TABU RESTRICTIONS	21
3.3.4	ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ SIMULATED ANNEALING	22
3.3.5	ΠΑΡΑΔΕΙΓΜΑ ONE POINT Crossover	24
3.3.6	ΠΑΡΑΔΕΙΓΜΑ TWO POINT Crossover	25
3.3.7	ΠΑΡΑΔΕΙΓΜΑ BIT - FLIP MUTATION	25
3.3.8	VNS BASIC	26
3.3.9	VNS ALGORITHM	27
3.3.10	ND ALGORITHM.	28
3.3.11	REDUCED VNS ALGORITHM	28

3.3.12 ■HAKE VNS ALGORITHM	28
4.2.1 DRONE BASED PARCEL DELIVERY USING THE ROOFTOPS OF THE BUILDINGS	31
4.2.2 EXAMPLE DRONE SPECIFICATIONS	32
5.3.1 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ	54
5.3.2 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ	55
5.3.3 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ	56

Κατάλογος Πινάκων

4.1	ΣΥΝΟΛΙΚΟ ΜΕΓΕΘΟΣ ΤΩΝ ΠΑΚΕΤΩΝ (TOTAL LENGTH OF BLOCKS)	34
4.2	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1	34
4.3	ΧΡΟΝΟΣ ΦΟΡΤΙΣΗΣ ΓΙΑ ΤΗΝ ΜΕΤΑΦΟΡΑ ΠΑΚΕΤΩΝ (<i>CHARGING TIME OF DRONES</i>)	36
4.4	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 2	36
4.5	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 3	38
4.6	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1	40
4.7	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 1	40
4.8	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1	42
4.9	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 2	42
4.10	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1	45
4.11	ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 3	45
5.1	ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ	48
5.2	ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ	49
5.3	ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ	50
5.4	ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ	51
5.5	ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ	52
5.6	ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ	53

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 ΠΡΟΒΛΗΜΑ - ΣΗΜΑΝΤΙΚΟΤΗΤΑ ΤΟΥ ΘΕΜΑΤΟΣ

Το θέμα που πραγματεύεται η παρούσα διπλωματική εργασία είναι τα προβλήματα δρομολόγησης. Η βελτιστοποίηση της διαδικασίας μεταφοράς προϊόντων είναι κάτι σημαντικό για μια επιχείρηση. Εύκολα κάποιος μπορεί να καταλάβει ότι με την μεγιστοποίηση του αριθμού των αντικειμένων που μεταφέρονται μέσα σε ένα καθορισμένο χρονικό διάστημα αποφέρει μεγαλύτερο κέρδος σε μία επιχείρηση. Αυτό το πρόβλημα έρχονται να λύσουν διάφοροι αλγόριθμοι δρομολόγησης οχημάτων. Τα VRP προβλήματα ανήκουν στην κατηγορία των NP - Hard προβλημάτων. Η παρούσα διπλωματική εργασία αφορά πρόβλημα δρομολόγησης με την χρήση drone.(Kim, Moon, and Jung 2020) Στο συγκεκριμένο πρόβλημα έχουμε έναν αριθμό από πακέτα τα οποία πρέπει να παραδοθούν σε πελάτες με χρήση drones τα οποία ξεκινούν και ολοκληρώνουν την διαδρομή τους από μία αφετηρία (depot). Ο στόχος είναι να μεταφερθούν όσο περισσότερα πακέτα γίνεται μέσα σε 9 ώρες. Το πρόβλημα λύνεται αρχικά με την κατασκευή μίας αρχικής λύσης με την χρήση ενός εκ των τριών αλγορίθμων που παρουσιάζονται. Τέλος για την επίτευξη τελικής λύσης χρησιμοποιείται η μεθευρετική μέθοδος (VNS).

1.2 ΣΚΟΠΟΣ ΕΡΓΑΣΙΑΣ

Η παρούσα εργασία έχει ως σκοπό την διερεύνηση και παρουσίαση μεθευρετικών μεθόδων για την επίλυση ενός προβλήματος της κατηγορίας VRP. Η παρούσα θέση εστιάζεται στο πρόβλημα μεταφοράς αντικειμένων με την χρήση drone από ένα κεντρικό κτίριο (Drone Operation Center) σε διάφορα κτίρια μέσα στην πόλη με την χρήση της οροφής του κάθε κτιρίου ως σημείου παράδοσης αντικειμένων. Η εργασία έχει ως αρχικό στόχο την κατανόηση του προβλήματος μέσα από μία βιβλιογραφική ανασκόπηση. Στην συνέχεια παρουσιάζονται τρεις αλγόριθμοι κατασκευής αρχικής λύσης και τρεις αλγόριθμοι βελτίωσης με απώτερο σκοπό την εύρεση όσο το δυνατόν καλύτερης αρχικής λύσης και βελτίωσης αυτής με την χρήση της Variable Neighborhood Search μεθευρετικής

στην γλώσσα προγραμματισμού python. Τέλος θα πραγματοποιηθεί μία συγκριτική υπολογιστική μελέτη εφαρμογής των αλγορίθμων κατασκευής αρχικής λύσης και της VNS μεθευρετικής πάνω σε διαθέσιμα προβλήματα και θα αξιολογεί την αποτελεσματικότητά τους ως προς τον αριθμό των πακέτων που περισσεύουν.

1.3 ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΜΕΛΕΤΗΣ

Στην παρούσα υποενότητα περιέχεται η δομή με την οποία έγινε η ανάπτυξη της παρούσας διπλωματικής εργασίας. Η οποία συνοπτικά περιέχει έξι κεφάλαια. Στο πρώτο κεφάλαιο το οποίο είναι εισαγωγικό, παρουσιάζονται στον αναγνώστη οι στόχοι της διπλωματικής εργασίας και γίνεται περιληπτική αναφορά στη διάρθρωση της εργασίας. Στην συνέχεια έχουμε το δεύτερο κεφάλαιο στο οποίο παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο με σκοπό την ενημέρωση του αναγνώστη και όσο το δυνατό καλύτερη κατανόηση του θέματος πάνω στο πρόβλημα που παρουσιάζεται. Στην συνέχεια έχουμε το τρίτο κεφάλαιο στο οποίο γίνεται μία εκτενής αναφορά στους αλγόριθμους επίλυσης προβλημάτων που ανήκουν στην κατηγορία VRP. Στο τέταρτο κεφάλαιο γίνεται η περιγραφή του προβλήματος και παρουσιάζονται οι αλγόριθμοι που χρησιμοποιούνται για την επίλυση του καθώς και ένα παράδειγμα για τον κάθε αλγόριθμο. Έπειτα στο πέμπτο κεφάλαιο γίνεται η παρουσίαση των αποτελεσμάτων και σύγκριση αυτών μετά την χρήση των αλγορίθμων. Τέλος στο έκτο και τελευταίο κεφάλαιο έχουμε τα συμπεράσματα που προέκυψαν και τις μελλοντικές επεκτάσεις της εργασίας.

1.4 ΣΥΝΕΙΣΦΟΡΑ

Στην παρούσα εργασία γίνεται η παρουσίαση του προβλήματος του πλανόδιου πωλητή, όπως και του προβλήματος δρομολόγησης οχημάτων και αλγορίθμων επίλυσης του. Επίσης παρουσιάζεται πρόβλημα μεταφοράς αντικειμένων με την χρήση drone. Αφού το πρόβλημα που λύνεται αφορά ειδικά τα drones για αυτό τον λόγο γίνεται και μία αναφορά σε αυτά.

Η ουσιαστική συνεισφορά της παρούσας εργασίας συνοψίζεται στα ακόλουθα

- Δύο νέα construction heuristics για την επίλυση του προβλήματος μεταφοράς αντικειμένων με την χρήση drone.
- Δύο νέα improvement heuristics για την επίλυση του προβλήματος μεταφοράς αντικειμένων με την χρήση drone.

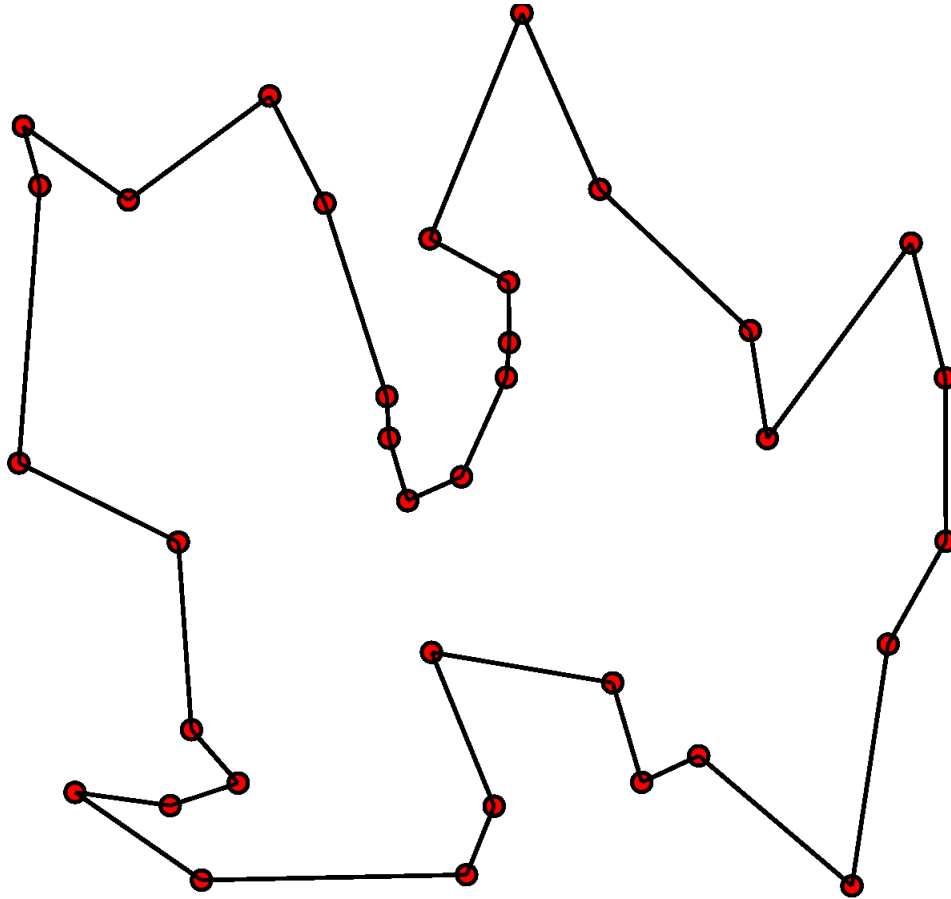
ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

2.1 ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ

Το πρόβλημα του πλανόδιου πωλητή είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης. (Jünger, Reinelt, and Rinaldi 1995) Έχοντας μία λίστα από πόλεις και τις αποστάσεις από και προς στην κάθε μία, έχουμε στόχο την εύρεση της πιο σύντομης διαδρομής έχοντας επισκεφτεί την κάθε πόλη ακριβώς μία φορά. Το πρόβλημα για πρώτη φορά μοντελοποιήθηκε μαθηματικά το 1930. Είναι ένα πρόβλημα που έχει απασχολήσει πολλές φορές τους ερευνητές. Χρησιμοποιείται σαν παράδειγμα για πολλές μεθόδους βελτιστοποίησης. Το πρόβλημα του πλανόδιου πωλητή εφαρμόζεται σε προβλήματα όπως (π.χ. προβλήματα μεταφοράς και δημιουργίας microchips). Ακόμα έχει εφαρμογή και σε προβλήματα ακολουθίας DNA. Σε τέτοιες περιπτώσεις οι πόλεις παρουσιάζονται σαν πελάτες, κομβικά σημεία ή τμήματα DNA, η απόσταση παρουσιάζεται σαν χρόνο ταξιδιού ή κόστος ή μέθοδος ομοιότητας μεταξύ των τμημάτων DNA. Σε πολλές εφαρμογές υπάρχουν και άλλοι περιορισμοί όπως χρονικά παράθυρα που δυσκολεύουν το πρόβλημα. Όσον αφορά την υπολογιστική πολυπλοκότητα το πρόβλημα του πλανόδιου πωλητή κατατάσσεται στα NP - Complete προβλήματα.

https://en.wikipedia.org/wiki/Travelling_salesman_problem



ΣΧΗΜΑ 2.1.1: TSP

2.1.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ιστορικά δεν είναι σαφές από που προέρχεται το πρόβλημα του πλανόδιου πωλητή. Το 1832 σε ένα εγχειρίδιο γίνεται μία αναφορά στο πρόβλημα περιέχονται επίσης μερικά παραδείγματα διαδρομών διαμέσου Γερμανίας και Σουηδίας αλλά χωρίς κάποια μαθηματική υποστήριξη. Σχετιζόμενα προβλήματα με το πρόβλημα του πλανόδιου πωλητή αντιμετωπίστηκαν από τους μαθηματικούς W. R. Hamilton και Thomas Kirkman. Η γενική μορφή του προβλήματος του πλανόδιου πωλητή φαίνεται να πρωτομελετήθηκε από μαθηματικούς κατά την διάρκεια της δεκαετίας του 1930 στην Βιέννη και το Χάρβαρντ, ιδιαίτερα από τον Karl Menger ο οποίος ορίζει το πρόβλημα και παρατηρεί την μη-βέλτιστη λύση της nearest neighbor ευρετικής. Λίγο αργότερα στο πανεπιστήμιο του Princeton του δόθηκε η ονομασία Travelling salesman problem από τον Hassler Whitney (Schrijver 2005). Την δεκαετία του 1950 και του 1960 το πρόβλημα έγινε αρκετά δημοφιλές στην Αμερική και στην Ευρώπη. Αξιοσημείωτες συνεισφορές έγιναν από τους George Dantzing, Delbert Ray Fulkerson και Selmer M. Johnson στην εταιρία RAND από την Santa Monica όπου το εξέφρασαν σαν ακέραιο γραμμικό πρόβλημα και δημιούργησαν την cutting plane μέθοδο για την λύση του. Με αυτές τις

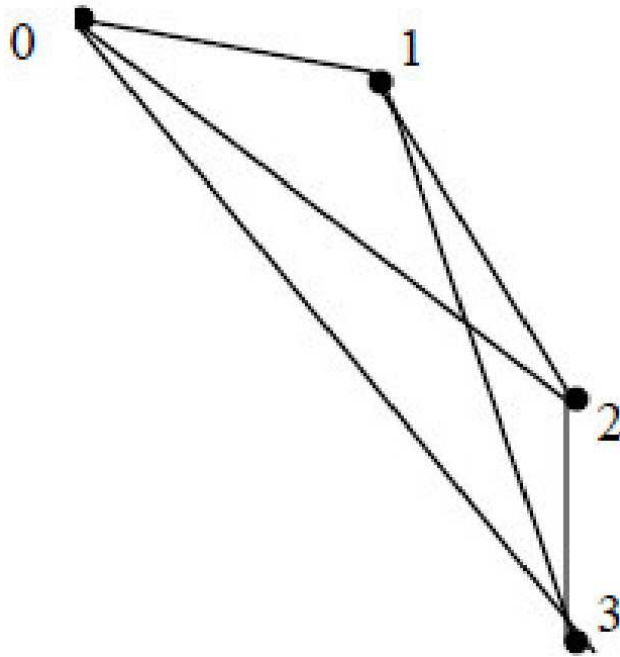
νέες μεθόδους κατάφεραν και έλυσαν βέλτιστα ένα παράδειγμα με 49 πόλεις έχοντας δημιουργήσει μία διαδρομή και αποδεικνύοντας ότι δεν μπορεί να υπάρξει συντομότερη. Στις επόμενες δεκαετίες το πρόβλημα μελετήθηκε από πολλούς ερευνητές των μαθηματικών, της επιστήμης των υπολογιστών, της χημείας, της φυσικής και άλλων επιστημονικών πεδίων. Το 1972 ο Richard M. Karp έδειξε ότι το πρόβλημα Χαμιλτονιανού κύκλου ανήκει στην κατηγορία NP - complete, όπου αυτό υποδηλώνει την NP - δυσεπιλυσιμότητα του TSP. Αυτό περιείχε και μία μαθηματική επεξήγηση για την προφανή υπολογιστική δυσκολία εύρεσης βέλτιστων διαδρομών. Μεγάλη πρόοδος έγινε από τους Grötschel, Padberg, Rinaldi και άλλους όταν αυτοί στα τέλη της δεκαετίας του 1970 - 1980 κατάφεραν να λύσουν βέλτιστα προβλήματα που είχαν μέχρι 2392 πόλεις χρησιμοποιώντας τις τεχνικές cutting planes και branch-and-bound. Στην δεκαετία του 1990 οι Applegate, Bixby, Chvátal, και Cook δημιούργησαν έναν λύτη του προβλήματος το πρόγραμμα Concorde. Ο Gerhard Reinelt το 1991 δημοσίευσε την TSPLIB μία συλλογή από benchmark προβλήματα ποικίλης δυσκολίας η οποία χρησιμοποιείται κατά καιρούς από ερευνητές για σύγκριση αποτελεσμάτων. (Reinelt 1991) Το 2006 ο Cook με την ομάδα του υπολόγισαν την βέλτιστη διαδρομή για ένα παράδειγμα 85900 πόλεων.

2.1.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Το TSP μπορεί να μοντελοποιηθεί σαν γράφος, όπου οι πόλεις είναι οι κορφές του γράφου, οι ακμές είναι τα μονοπάτια και η απόσταση του μονοπατιού είναι το μήκος των ακμών. Μία TSP διαδρομή γίνεται ένας Χαμιλτονιανός κύκλος και η βέλτιστη διαδρομή είναι ο μικρότερος Χαμιλτονιανός κύκλος.

2.1.3 ΣΥΜΜΕΤΡΙΚΟ TSP

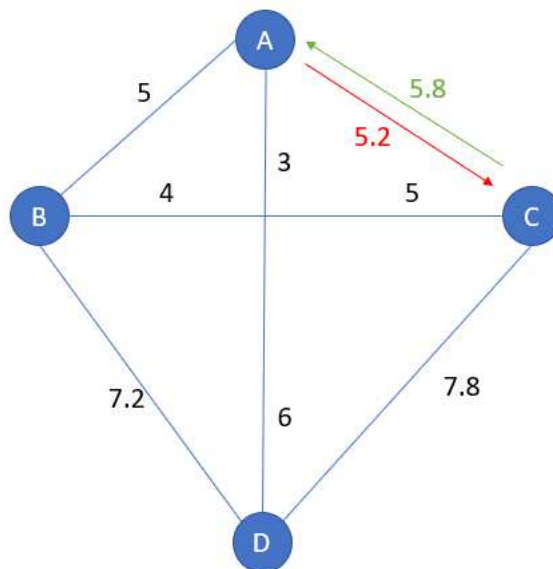
Στο συμμετρικό TSP η απόσταση μεταξύ δύο πόλεων είναι η ίδια και από τις δύο κατευθύνσεις σχηματίζοντας έναν μη κατευθυνόμενο γράφο (Chaudhuri and De 2011).



ΣΧΗΜΑ 2.1.2: SYMETRIC TSP

2.1.4 ΜΗ ΣΥΜΜΕΤΡΙΚΟ TSP

Στο μη συμμετρικό TSP μπορεί να μην υπάρχουν μονοπάτια και από τις δύο κατευθύνσεις ή οι αποστάσεις μπορεί να μην είναι ίδιες σχηματίζοντας έναν κατευθυνόμενο γράφο. <https://datascience.lc/2020/03/01/travelling-salesman-problem-tsp-concept/>



ΣΧΗΜΑ 2.1.3: ASYMETRIC TSP

2.1.5 ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ TSP

Data:	
$I = \{1, \dots, n\}$: set of cities	
$h_{i,j}, \forall (i,j) \in I \times I$: cost necessary to travel from the city i to the city j	
Variables:	
$x_{i,j} = \begin{cases} 1 & \text{if the city } i \text{ precedes the city } j \\ 0 & \text{otherwise} \end{cases}, \forall (i,j) \in I \times I$	
Minimize:	
$\sum_{i \in I} \sum_{j \in I} h_{i,j} \times x_{i,j}$	[2.14]
Under the Constraints:	
$\sum_{j \in I} x_{i,j} = 1, \forall i \in I$	[2.15]
$\sum_{i \in I} x_{i,j} = 1, \forall j \in I$	[2.16]
$\sum_{i \in S} \sum_{j \in S} x_{i,j} \leq \text{Card}(S) - 1, \forall S \subset I, 1 \leq \text{Card}(S) \leq \left\lfloor \frac{n}{2} \right\rfloor$	[2.17]

ΣΧΗΜΑ 2.1.4: TSP MATHEMATICAL MODEL

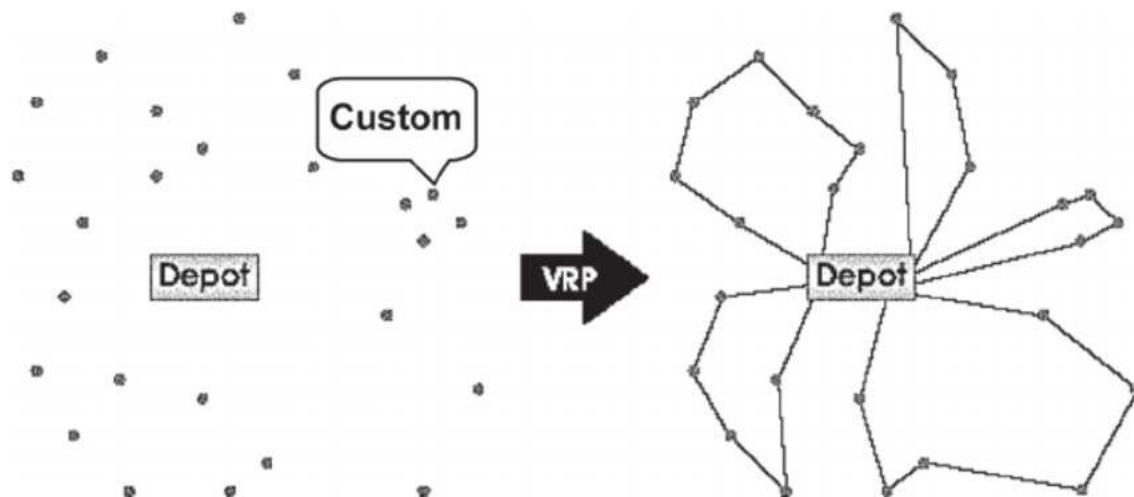
Στην παραπάνω εικόνα 2.1.4 παρουσιάζεται η μαθηματική μοντελοποίηση του TSP. (Deroussi 2016b) Έχοντας ως αντικειμενική συνάρτηση [2.14] και περιορισμούς [2.15] και [2.16] προσπαθούμε να καταναείμουμε σε κάθε πόλη την πόλη που θα ακολουθήσει σύμφωνα με την καθορισμένη διαδρομή. Σε αυτό το μοντέλο $h_{i,j}$ αντιπροσωπεύει το κόστος (απόάποψη απόστασης) που είναι απαραίτητο να διανύσουμε από την πόλη i έως την πόλη j . Με την χρήση του περιορισμού [2.17] εξαλείφουμε τους υποκύκλους που προκύπτουν έτσι αν έχουμε $\Sigma=1,3$ δεν επιτρέπεται να έχουμε τα τόξα (1,3) και (3,1) στην λύση μας και έτσι αποτρέπουμε τον υποκύκλο 1-3-1.

2.2 VRP

2.2.1 ΕΙΣΑΓΩΓΗ

Η δρομολόγηση οχημάτων είναι ένα κλασικό πρόβλημα επιχειρησιακής έρευνας, ανήκει στην κατηγορία των NP - Hard προβλημάτων και στην βιβλιογραφία το συναντάμε με το όνομα Vehicle Routing Problem (VRP) ή Capacitated VRP (CVRP). Καθορίζεται από ένα χαμηλότερο κόστους σύνολο διαδρομών από μία αποθήκη (depot) για ένα στόλο από οχήματα με χωρητικότητα με σκοπό να ικανοποιήσουν τις απαιτήσεις από ένα σύνολο πελατών. Η έρευνα σε εφαρμογές που

σχετίζονται με το CVRP και τις παραλλαγές τους το έχουν κάνει ένα από τα πιο μελετημένα προβλήματα συνδυαστικής βελτιστοποίησης. Το 1959 οι Dantzing και Ramser (Dantzig and Ramser 1959) παρουσίασαν το CVRP με το όνομα truck dispatching problem για να μοντελοποιήσουν ένα πραγματικό πρόβλημα διανομής βενζίνης σε πρατήρια καυσίμων. Μετά από αυτό τα μοντέλα και οι μέθοδοι λύσεων για το CVRP και τις παραλλαγές του είχαν μεγάλη αύξηση. Το 2009 οι Eksioglu et al (Eksioglu, Vural, and Reisman 2009) κατηγοριοποίησαν περισσότερα από 1000 άρθρα. Το 2009 ο Laporte (Laporte 2009) με ένα άρθρο του έκανε ουσιαστικά μία περίληψη των επιτευγμάτων στα 50 χρόνια του προβλήματος. Παρόλο όμως που υπάρχει μεγάλη δραστηριότητα μέχρι πρότινος μπορούσαν να βρουν βέλτιστη λύση σε προβλήματα μέχρι 100 πελατών (Baldacci, Christofides, and Mingozzi 2008) και πρόσφατα καταφέρνουν τους 200 (Pessoa, Sadykov, and Uchoa 2017) αλλά όμως σε πραγματικές περιπτώσεις μπορεί να υπάρχουν περισσότεροι από 1000 πελάτες. (Siarry 2016) (Pourrahmani, Delavar, and Mostafavi 2015)



ΣΧΗΜΑ 2.2.5: CLASSIC - VEHICLE - ROUTING - PROBLE (VRP)

2.2.2 ΒΑΣΙΚΗ ΜΟΡΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

Το βασικό CVRP ορίζεται σαν ένα complete μη κατευθυνόμενο γράφημα $G = (V, E)$. Με ένα σύνολο V κόμβων όπου περιέχεται και μία αποθήκη (depot, κόμβος 0), όπου υπάρχει ένας στόλος οχημάτων χωρητικότητας Q και με πελάτες που απαιτούν q_i για ένα προϊόν όπου $i = 1, 2..n$. Κάθε ακμή $[i, j]$ στο σύνολο e αντιπροσωπεύει ένα βέλτιστο μονοπάτι ανάμεσα στους κόμβους i, j στο οδικό δίκτυο. Το κόστος c_{ij} η απόσταση ή ο χρόνος ταξιδιού είναι συχνά προυπολογισμένα. Ο σκοπός είναι να καθορισθεί ένα σύνολο από διαδρομές με το λιγότερο συνολικό κόστος έχοντας επισκεφτεί όλους τους πελάτες ακριβώς μία φορά. Μία διαδρομή είναι ένας κύκλος που ξεκινάει και τελειώνει στην αποθήκη (depot), που πραγματοποιείται από ένα όχημα και το συνολικό βάρος δεν ξεπερνάει το Q . Από τους συγγραφείς του προβλήματος εξαρτάται ο αριθμός των οχημάτων ο οποίος μπορεί να είναι σταθερός ή ελεύθερος, ο χρόνος εξυπηρέτησης s_i για κάθε πελάτη μερικές

φορές είναι καθορισμένος και οι διαδρομές μπορεί να περιοριστούν από την μέγιστη απόσταση ή από χρόνο ταξιδιού L . Παρακάτω **2.2.6** έχουμε ένα απλό μοντέλο, όπου έχουμε δύο κόμβους σαν depot όπου χρησιμοποιούνται για την αρχή και το τέλος κάθε διαδρομής, κάθε ακμή $[i, j]$ αντικαθίσταται από τα δύο αντίθετα τόξα (i, j) και (j, i) . Η δυαδική μεταβλητή x είναι ίση με 1 αν το τόξο (i, j) διασχίζεται από το όχημα k .

$$\min \sum_k \sum_{(i,j)} c_{ij} \cdot x_{ij}^k \quad (15.1)$$

$$\sum_{j \neq i} \sum_k x_{ij}^k = 1 \quad \forall i \neq 0, n+1 \quad (15.2)$$

$$\sum_{j \neq i} x_{ji}^k = \sum_{j \neq i} x_{ij}^k \quad \forall i \neq 0, n+1 \quad \forall k \quad (15.3)$$

$$\sum_{i \neq 0, n+1} \sum_{j \neq i} q_i \cdot x_{ij}^k \leq Q \quad \forall k \quad (15.4)$$

$$t_i^k + s_i + c_{ij} \leq t_j^k + M(1 - x_{ij}^k) \quad \forall i \quad (15.5)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \quad \forall k \quad (15.6)$$

$$t_i^k \geq 0 \quad \forall i \quad \forall k \quad (15.7)$$

ΣΧΗΜΑ 2.2.6: VRP MODEL

Το πρόβλημα που παρουσιάζεται στην παραπάνω εικόνα **2.2.6** είναι πρόβλημα ελαχιστοποίησης. (Siarry 2016) Αυτό φαίνεται από την αντικειμενική συνάρτηση στην οποία πρέπει να ελαχιστοποιηθεί το κόστος των διαδρομών. Ο περιορισμός 15.2 μας εγγυάται ότι θα ικανοποιηθούν όλοι οι πελάτες. Ο περιορισμός 15.3 εγγυάται ότι το όχημα που φτάνει στον πελάτη είναι αυτό που φεύγει από τον πελάτη. Ο περιορισμός 15.4 αναπαριστά την χωρητικότητα του οχήματος. Η εξίσωση 15.5 αποτρέπει τους υποκύκλους.

2.2.3 ΠΑΡΑΛΛΑΓΕΣ ΚΛΑΣΣΙΚΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Παρόλο που το κλασσικό πρόβλημα VRP είναι πολύ δημοφιλές, υπάρχουν πολλές παραλλαγές του προβλήματος. Αρχικά υπάρχουν παραλλαγές που με επιπρόσθετα χαρακτηριστικά ή περιορισμούς που μπορεί να επηρεάσουν τους πελάτες. (Talbi 2009) (Nazif and Lee 2012) Αυτές είναι:

- Vehicle Routing Problem with Time Windows (VRPTW) στο οποίο οι απαιτήσεις των πελατών πρέπει να καλυφθούν μέσα σε συγκεκριμένο χρόνο. (Ghannadpour and Hooshfar 2016) (Lei, Laporte, and Guo 2011) (Hong 2012) (Vidal, Crainic, Gendreau, and Prins

2013)

- Pickup and Delivery Problems (PDPs) όπου έχουν ως στόχο την ολοκλήρωση ενός συνόλου αιτημάτων για μεταφορά από έναν κόμβο παραλαβής σε έναν κόμβο μεταφοράς εν αντιθέσει με το κλασικό CVRP όπου όλα τα αντικείμενα κατανέμονται από το depot. (D'Souza, S N, and Senthilnath 2012) (Goksal, Karaoglan, and Altiparmak 2013) (Qu and Bard 2012)
- Το Dial-a-Ride Problem όπου έχουν ως στόχο τον σχεδιασμό διαδρομών και δρομολογίων οχημάτων για πελάτες που καθορίζουν αιτήματα παραλαβής και παράδοσης μεταξύ προέλευσης και προορισμού. (Parragh and Schmid 2013) (Cordeau and Laporte 2007)
- Το Team Orienteering Problem (TOP) με σκοπό να προσδιοριστούν σύνολα από διαδρομές οι οποίες μεγιστοποιούν το μεγαλύτερο κέρδος, δεδομένου το ότι έχουμε περιορισμένο χρόνο να ικανοποιήσουμε όλους τους πελάτες. (Labadie, Mansini, Melechovský, and Calvo 2012) (Lin 2013) (Lin and Yu 2012)

Επίσης υπάρχουν παραλλαγές που σχετίζονται με τα οχήματα. Όπως:

- Το Heterogeneous Fleet VRP (HFVRP) το οποίο εξετάζει διάφορους τύπους οχημάτων ανάλογα με την διαθεσιμότητα και συγκεκριμένο κόστος ανά την απόσταση. (Brandão 2011) (Duhamel, Lacomme, and Prodhon 2012) (Naji-Azimi and Salari 2013)
- Τα Truck and Trailer Routing Problems (TTRPs) τα οποία περιέχουν πολύπλοκα δρομολόγια, όπου κάθε truck μπορεί προσωρινά να αφήσει το trailer σε κάποιον πελάτη για να μπορέσει να επισκεφτεί περιοχές που δεν είναι δυνατό όταν το όχημα είναι ολόκληρο. (Villegas, Prins, Prodhon, Medaglia, and Velasco 2011) (Lin, Yu, and Lu 2011)

Εκτός των άλλων υπάρχουν επεκτάσεις που σχετίζονται με τον τύπο διαδρομών, τον σχεδιασμό και τα κριτήρια βελτιστοποίησης. Όπως:

- Τα Multi Depot VRP (MDVRP) στα οποία υπάρχουν περισσότερα από ένα depots. (Aras, Aksen, and Tuğrul Tekin 2011) (Kuo and Wang 2012)
- Τα Open VRP (OVRP) όπου τα οχήματα δεν είναι απαραίτητο να επιστρέψουν στο depot μετά την ολοκλήρωση της υπηρεσίας τους (Li, Leung, and Tian 2012).
- Τα Periodic Vehicle Routing Problems (PVRP), όπου τα οχήματα πρέπει να επισκεφτούν τον κάθε πελάτη με μία συγκεκριμένη συχνότητα (Yu and Yang 2011). (Cacchiani, Hemmelmayr, and Tricoire 2014)

2.3 DRONES

Το Unmanned Aerial Vehicle ή αλλιώς drones είναι ουσιαστικά αεροπλάνο χωρίς πιλότο. <https://www.theuav.com/> Τα drones μπορεί κάποιος να τα πιλοτάρει απομακρυσμένα. Ακόμα υπάρχει η δυνατότητα τα drones να πετάζουν αυτόματα βασισμένα σε προγραμματισμένες πτήσεις. <https://www.pcmag.com/encyclopedia/term/uav> Τα τελευταία χρόνια τα drones έχουν δείξει ότι υπάρχει δυνατότητα χρήσης τους για μεταφορά πακέτων. Η μεταφορά με χρήση drones φαίνεται να είναι πιο γρήγορη, λιγότερο ακριβή και πιο φιλική προς το περιβάλλον. Τα εμπορικά drones λειτουργούν με μπαταρίες. Έχουν διεξαχθεί πολλές ακαδημαϊκές έρευνες που σχετίζονται με την μεταφορά βασισμένη σε drones (drone - based delivery). Οι (Ha, Deville, Pham, and H 2018) πρότειναν ένα TSP με drones το οποίο έχει ως σκοπό την ελαχιστοποίηση του κόστους μεταφοράς συγκριτικά με φορτηγά. Περισσότερα παραδείγματα παρουσιάζονται στο κεφάλαιο 3 όπου γίνεται αναλυτική αναφορά στους αλγόριθμους επίλυσης προβλημάτων κατηγορίας VRP.



ΣΧΗΜΑ 2.3.7: ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΑΕΡΟΣΚΑΦΗ

2.3.1 ΚΑΤΗΓΟΡΙΕΣ DRONES

Τρεις είναι οι κύριες κατηγορίες drone τα οποία είναι διαθέσιμα για χρήση από τους πολίτες. <https://www.baatraining.com/uav-types-how-to-choose-yours/> Οι κατηγορίες είναι:

- Σταθερής πτέρυγας (Fixed - Wing)
- Περιστροφικής πτέρυγας (Rotary - Wing)
- Πολλαπλών ελίκων (Multi - Rotor)

ΣΤΑΘΕΡΗΣ ΠΤΕΡΥΓΑΣ (FIXED - WING)

Τα drones σταθερής πτέρυγας αποτελούνται από σταθερές πτέρυγες πάνω στον κορμό του. Είναι ικανά να κουβαλήσουν περισσότερο εξοπλισμό ή βάρος σε μεγαλύτερες αποστάσεις και με λιγότερη ενέργεια. Μπορούν επίσης να χρησιμοποιηθούν και σε υπηρεσίες παράδοσης πακέτων. Αν όμως μας ενδιαφέρει η ακρίβεια θα πρέπει να χρησιμοποιήσουμε άλλο τύπο drone διότι λόγω της κατασκευής τους δεν μπορούν να αιωρηθούν σε ένα σημείο. <https://sentera.com/sentera-phoenix-2-uav-launch/>



ΣΧΗΜΑ 2.3.8: FIXED WING

ΠΑΡΑΔΕΙΓΜΑ ΣΤΑΘΕΡΗΣ ΠΤΕΡΥΓΑΣ KWT-GM360-H

Το drone KWT-GM360-H ανήκει στην κατηγορία των drone σταθερής πτέρυγας. <http://www.alltechuav.com/>



ΣΧΗΜΑ 2.3.9: FIXED WING DRONE EXAMPLE

Κατασκευάζεται από την εταιρία KEWEITAI που εδρεύει στην κίνα. Είναι εύκολο στην συναρμολόγηση του. Με σημαντικότερα χαρακτηριστικά την δυνατότητα κάθετης προσγείωσης και απογείωσης και άνοιγμα φτερών 3600mm. Πιο αναλυτικά τα χαρακτηριστικά του παρουσιάζονται στην παρακάτω εικόνα 2.3.10. http://www.alltechuav.com/product_detail.html?productId=79

Wingspan	3600mm
Fuselage length	2300mm
Body height	460mm
Maximum measurement and control distance	40km(LOS)
Runway	5m×5m
Takeoff/landing	vertical
Maximum takeoff weight	25kg
Power system	Petrol Engine/High efficiency brushless motor
Control mode	Fully autonomous control
Maximum wind tolerance	8.0-10.7m/s
Maximum endurance	4~6h@Petrol Engine, 2h@High efficiency brushless motor
Maximum level flight speed	110km/h@Petrol Engine,100km/h@High efficiency brushless motor
Maximum working load	5Kg@Petrol Engine, 3Kg@High efficiency brushless motor
Takeoff and landing mode	70~100km/h
Lift limit	4500m

ΣΧΗΜΑ 2.3.10: FIXED WING EXAMPLE DRONE CHARACTERISTICS

ΠΕΡΙΣΤΡΟΦΙΚΗΣ ΠΤΕΡΥΓΑΣ (ROTARY - WING)

Αυτός ο τύπος UAV μπορούμε να πούμε ότι είναι μία μικρότερη έκδοση του συμβατικού ελικοπτέρου αφού έχει έναν έλικα. Έχει την ικανότητα της κάθετης απογείωσης και προσγείωσης, το οποίο δίνει την δυνατότητα στον χρήστη να το χειρίζεται ακόμα και σε μικρούς χώρους. Η ικανότητα της αιώρησης και της εκτέλεσης ελιγμών καθιστά τα UAV περιστροφικής πτέρυγας κατάλληλα για εργασίες επιθεώρησης όπου οι ακριβείς ελιγμοί είναι αναγκαίοι. Μπορεί να χρησιμοποιηθεί στη επιθεώρηση σιδηροδρόμων, ηλεκτρικά καλώδια, γέφυρες και άλλα. Από την άλλη μεριά, αυτού του τύπου τα drones απαιτούν περισσότερη και πιο πολύπλοκη συντήρηση και διαδικασία επισκευής. <https://www.agriexpo.online/prod/alpha-unmanned-systems/product-176828-59132.html>

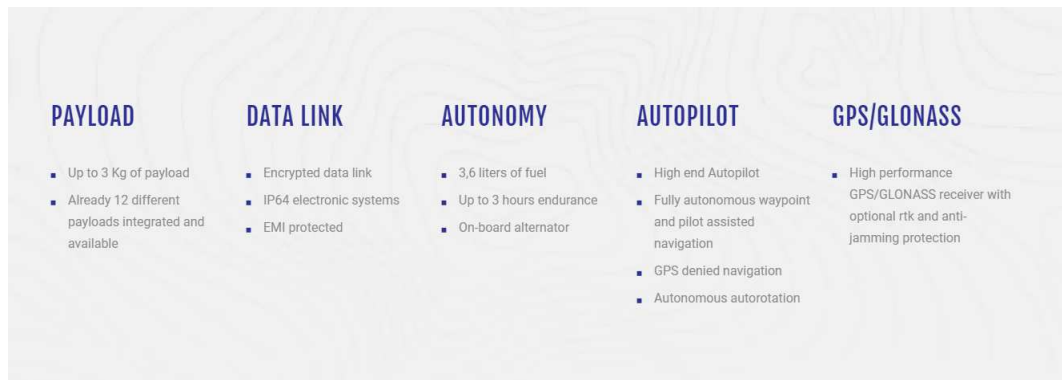
ΠΑΡΑΔΕΙΓΜΑ DRONE ΠΕΡΙΣΤΡΟΦΙΚΗΣ ΠΤΕΡΥΓΑΣ ALPHA 800



ΣΧΗΜΑ 2.3.11: ROTERY WING DRONE EXAMPLE

Το alpha 800 είναι ένα στρατιωτικό UAV από την alpha security and defence. Χρησιμοποιείται κυρίως για έλεγχο των συνόρων. Έχει αξιοσημείωτα χαρακτηριστικά όπως μέχρι και 3 ώρες διάρκεια πτήσης, μέχρι τα 50 χιλιόμετρα ακτίνα και δυνατότητα μεταφοράς φορτίου μέχρι

3 κιλά. Πιο αναλυτικά τα χαρακτηριστικά του παρουσιάζονται στην παρακάτω εικόνα **2.3.12**.
<https://alphasecurityanddefense.com/alpha-800/>



ΣΧΗΜΑ 2.3.12: ROTERY WING DRONE CHARACTERISTICS

ΠΟΛΛΑΠΛΩΝ ΕΛΙΚΩΝ (MULTI - ROTOR)

Τα drones της κατηγορίας πολλαπλών ελίκων έχουν τις ίδιες ιδιότητες με τα drones που ανήκουν στην κατηγορία περιστροφικής πτέρυγας, μόνο που είναι πιο εύκολα στον χειρισμό. Αυτό σημαίνει ότι μπορούν να χρησιμοποιηθούν σε περιπτώσεις που απαιτείται ακρίβεια στον χειρισμό. Τέτοιες αποστολές μπορεί να είναι η εναέρια βιντεοσκόπηση κάποιου γεγονότος ή χαρτογράφηση κάποιας περιοχής. Ένα σημαντικό μειονέκτημα των drones αυτής της κατηγορίας είναι ο περιορισμός όσο αφορά τον χρόνο πτήσης και την εμβέλεια πτήσης. Τα drones αυτά έχουν μικρή διάρκεια και εμβέλεια πτήσης και όλο αυτό έχει ως συνέπεια να μην μπορεί να ολοκληρωθεί ένα εγχείρημα σε μία μόνο πτήση και να χρειαστεί πολλές περισσότερες για την ολοκλήρωση του.

Το drones αυτής της κατηγορίας αποτελούνται από έναν αριθμό πτερύγων οι οποίες περιστρέφονται γύρω από έναν σταθερό ιστό. Οι πιο συνήθεις κατηγορίες είναι:

- τρικόπτερα
- τετρακόπτερα
- εξακόπτερα
- οκτακόπτερα

Εκτός από τα παραπάνω υπάρχουν και τα δωδεκόπτερα και δεκαεξακόπτερα που όμως είναι πιο σπάνια. http://www.heliceo.com/wp-content/uploads/2016/10/Heliceo_Fox4_multirotors_01-1.png



ΣΧΗΜΑ 2.3.13: MULRI-ROTOR

ΠΑΡΑΔΕΙΓΜΑ DRONE ΠΟΛΛΑΠΛΩΝ ΕΛΙΚΩΝ PD6B-AW-ARM DRONE



ΣΧΗΜΑ 2.3.14: PD6B - AW - ARM

Ανήκει στην κατηγορία των drones με πολλαπλούς έλικες. <https://www.prodrone.com/wpdir/wp-content/uploads/2018/04/PD6B-AW-ARM-01-scaled.jpg>

Distance between motors	1,620 mm / 63.8 inch
Height	800 mm / 31.5 inch
Weight	18.8 kg / 44.1 lb
Diameter of Propeller	685.8 mm / 27 inch
Flight Duration (Up to a blinking LED warning of low level)	10 - 20 min
Max. Payload	20 kg / 44.1 lb
Max Speed	60 km/h / 37.3 mph
Wind Speed	8 m/s / 17.9 mph
Water Resistance	All-weather type
Battery	16000mA x 2

ΣΧΗΜΑ 2.3.15: PD6B - AW - ARM DRONE CHARACTERISTICS

Το PD6B - AW - ARM κατασκευάζεται από την εταιρεία Prodrone της Ιαπωνίας. Οι δύο ρομποτικοί βραχίονες κάνουν αυτό το drone να ξεχωρίζει. Ανήκει στην κατηγορία των εξακόπτρων, μπορεί να κουβαλήσει μέχρι 18.8 kg και διάρκεια πτήσης 18 λεπτά. Πιο αναλυτικά τα χαρακτηριστικά του παρουσιάζονται στην παραπάνω εικόνα. **2.3.15** <https://www.prodrone.com/products/pd6b-aw-arm/>

ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΟΙ ΕΠΙΛΥΣΗΣ VRP

Για την επίλυση προβλημάτων της κατηγορίας CVRP έχουν χρησιμοποιηθεί κατά καιρούς διάφορες ευρετικές και μεθευρετικές μέθοδοι. (Laporte and Semet 2002) Οι κατηγορίες αυτών είναι:

- Μέθοδοι αρχικοποίησης (Construction heuristics)
- Μέθοδοι βελτίωσης (Improvement heuristics)
- Μεθευρετικές μέθοδοι (Metaheuristics)

3.1 ΜΕΘΟΔΟΙ ΑΡΧΙΚΟΠΟΙΗΣΗΣ

Τα construction heuristics χρησιμοποιούνται για την κατασκευή μίας λύσης από το μηδέν. Ένας από τους πιο διαδεδομένους construction heuristics είναι το Nearest Neighbor heuristic. (Gajpal, Roy, and Sahay 2019) Όπου σε CVRP προβλήματα το όχημα ξεκινά την διαδρομή από το depot και επισκέπτεται διαδοχικά τον πλησιέστερο πελάτη. Έτσι, σε κάθε επανάληψη, το όχημα μετακινείται από τον τωρινό πελάτη i στον πλησιέστερο πελάτη του j . Εάν η μετακίνηση στον πελάτη j υπερβαίνει τη χωρητικότητα του οχήματος Q , το όχημα επιστρέφει στο depot για να ξεκινήσει την νέα διαδρομή. Η επόμενη διαδρομή ξεκινά από την αποθήκη και επισκέπτεται τον πλησιέστερο πελάτη της αποθήκης. Με αυτόν τον τρόπο, δημιουργούνται διαφορετικές διαδρομές μέχρι να εξυπηρετηθούν όλοι οι πελάτες. Άλλα διαδεδομένα construction heuristics για την επίλυση CVRP προβλημάτων είναι (Avdoshin and Beresneva 2019):

- Sequential Insertion algorithm (SI)
- Improved Parallel Insertion algorithm
- Clarke and Wright Savings heuristic(CWS)

3.2 ΜΕΘΟΔΟΙ ΒΕΛΤΙΩΣΗΣ

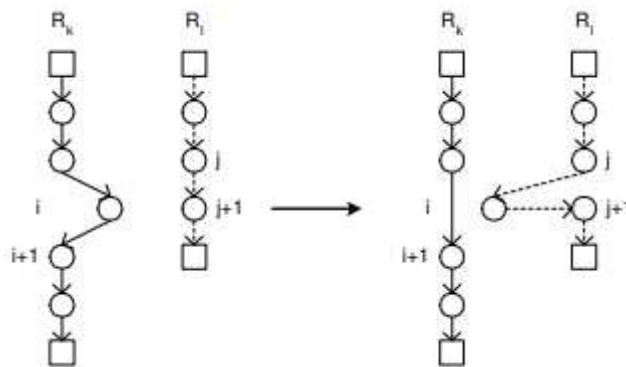
Μετά την εφαρμογή κάποιου construction heuristic σε ένα πρόβλημα της κατηγορίας VRP, μπορούμε να χρησιμοποιήσουμε κάποιο improvement heuristic για να βελτιώσουμε την αρχική λύση. Οι πιο κλασικοί αλγόριθμοι που χρησιμοποιούνται σε αυτού του είδους τα προβλήματα είναι (Ho and Haugland 2004):

- Relocate
- 2-opt

Για τα προβλήματα κατηγορίας VRP χωρίζουμε τα improvement heuristics σε intra - route και σε inter - route. Στην κατηγορία των intra - route γίνονται αλλαγές σε μία διαδρομή, ενώ στην inter - route γίνονται αλλαγές σε πολλές διαδρομές. (Tavares, Lopes, and Erig Lima 2009)

3.2.1 RELOCATE

Στην relocate μέθοδο γίνεται αφαίρεση ενός πελάτη από την κυκλική διαδρομή και την επανατοποθέτηση του στην θέση ενός άλλου πελάτη είτε στην ίδια κυκλική διαδρομή intra route είτε σε διαφορετική κυκλική διαδρομή inter - route. (Ho and Haugland 2004) (Gendreau and Potvin 2019)

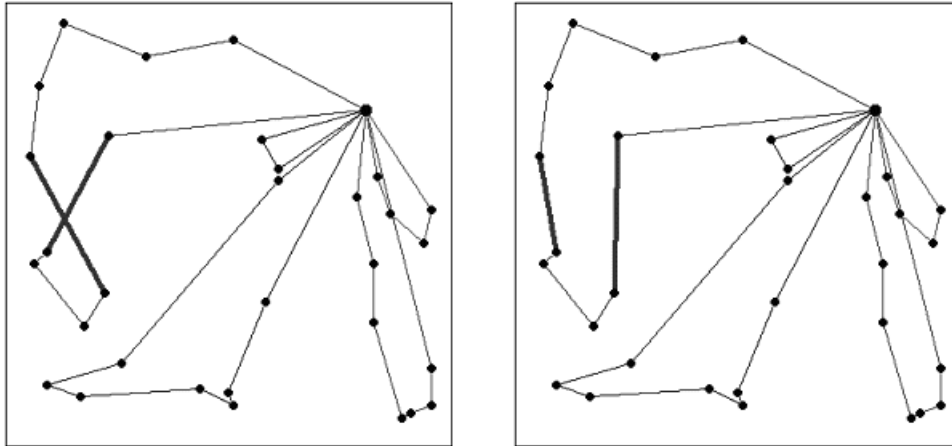


ΣΧΗΜΑ 3.2.1: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ RELOCATE INTER - ROUTE ΣΕ VRPTWSD ΠΡΟΒΛΗΜΑ

3.2.2 2-OPT

Ο Croes το 1958 πρότεινε τον 2-opt αλγόριθμο. (Croes 1958) Χρησιμοποιείται συχνά για την βελτίωση της αρχικής λύσης σε προβλήματα κατηγορίας VRP είτε σε intra - route είτε σε inter - route. Στην intra route περίπτωση αποκόπτονται δύο τυχαίες μη διαδοχικές ακμές από την ίδια κυκλική διαδρομή και ενώνονται μεταξύ τους για την δημιουργία νέας διαδρομής, ενώ στην

περίπτωση inter route οι ακμές που επιλέγονται πρέπει να ανήκουν σε διαφορετικές κυκλικές διαδρομές. (Tavares, Lopes, and Erig Lima 2009)



ΣΧΗΜΑ 3.2.2: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ 2 - OPT INTRA ROUTE ΣΕ CVRP

3.3 ΜΕΘΕΥΡΕΤΙΚΕΣ ΜΕΘΟΔΟΙ

Οι μεθευρετικές μέθοδοι είναι ένα αλγοριθμικό πλαίσιο υψηλού επιπέδου το οποίο παρέχει ένα σύνολο από κατευθύνσεις ή στρατηγικές για την δημιουργία ευρετικών αλγορίθμων βελτιστοποίησης. Ο όρος metaheuristic χρησιμοποιήθηκε πρώτη φορά από τον Glover (Glover 1986) συνδυάζοντας το πρόθεμα meta με την λέξη ευρετικός. Οι μεθευρετικές μέθοδοι μας παρέχουν αποδεκτές λύσεις σε μικρό χρονικό διάστημα για δύσκολα και περίπλοκα προβλήματα, δεν μπορούν όμως να μας εγγυηθούν την βέλτιστη λύση, ούτε να ορίσουν πόσο κοντά είμαστε από αυτή. Η αποδοτικότητα των μεθευρετικών μεθόδων έχει αυξήσει το ενδιαφέρον σε επιστήμονες και εταιρίες ώστε να τους χρησιμοποιούν σε προβλήματα βελτιστοποίησης του πραγματικού κόσμου. Μερικά παραδείγματα μεθευρετικών αλγορίθμων είναι η tabu search, simulated annealing, VNS και άλλοι. (Talbi 2009)

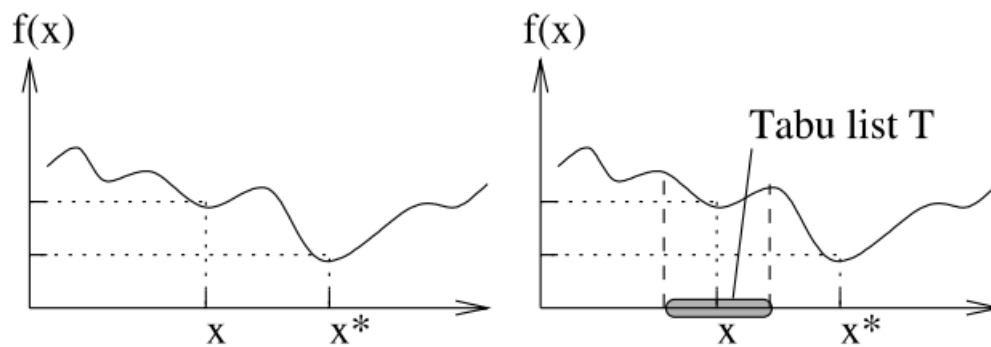
Η μεθευρετικές μεθοδολογίες (metaheuristics) συνήθως χωρίζονται σε δύο ευρείς κατηγορίες.(Deroussi 2016a) Οι κατηγορίες αυτές είναι:

- individual - based όπου αυτές βασιζόμενες σε ένα σύστημα γειτονιών κινούνται στον χώρο αναζήτησης.
- population - based οι οποίες βασίζονται σε έναν πληθυσμό λύσεων.

3.3.1 TABU SEARCH

Η tabu search μεθευρετική προτάθηκε από τον Glover.(Glover 1986) Η tabu search έχει ως στόχο να απεμπλακούμε από κάποιο τοπικό ελάχιστο. Για να το πετύχει αυτό η tabu search χρησιμοποιεί μία λίστα την tabu list. Στην tabu list αποθηκεύονται λύσεις που έχουν βρεθεί με

την χρήση τοπικής αναζήτησης. Λύσεις που είναι αποθηκευμένες στη tabu list δεν επιτρέπεται να ξαναχρησιμοποιηθούν, έτσι η tabu search για να προχωρήσει αποδέχεται καινούριες χειρότερες αντικειμενικά λύσεις από το ελάχιστο που έχουμε ήδη. Για να αποφευχθούν προβλήματα μνήμης πολλές φορές το μέγεθος της tabu list είναι περιορισμένο, έτσι όταν η λίστα γεμίσει από λύσεις τότε για να προστεθούν νέες βγαίνουν οι πιο παλιές λύσεις οι οποίες δεν θεωρούνται πια tabus. (Rothlauf 2011)



ΣΧΗΜΑ 3.3.3: TABU RESTRICTIONS

Τα πιο συχνά επιλεγμένα κριτήρια τερματισμού της tabu search είναι (Gendreau and Potvin 2019):

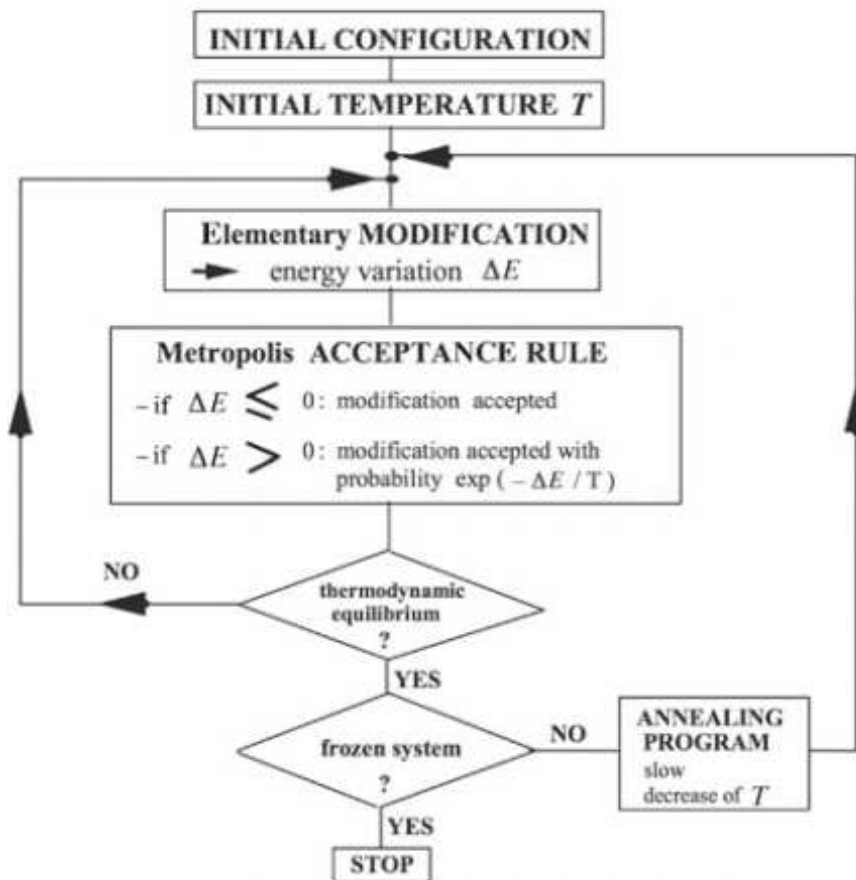
- Μετά από κάποιο χρονικό όριο CPUtime.
- Μετά από έναν αριθμό επαναλήψεων χωρίς βελτίωση της τιμής της αντικειμενικής συνάρτησης.

Στην δεκαετία του 1990 η tabu search μεθοδολογία ήταν η πιο αποτελεσματική για προβλήματα της κατηγορίας VRP. Υπάρχουν επίσης επιτυχημένες προσπάθειες σε προβλήματα των κατηγοριών CVRP (Liao 2004) (Rego and Roucairol 1996) (Toth and Vigo 2003), VRPTW (Cordeau, Laporte, and Mercier 2001), HFVRP (Brandão 2011).

3.3.2 SIMULATED ANNEALING

Η μεθευρετική Simulated Annealing εμφανίστηκε στις αρχές της δεκαετίας 1980. (Kirkpatrick, Gelatt, and Vecchi 1983) Η μεθευρετική Simulated Annealing βασίζεται στη μίμηση της διαδικασίας της απόπτησης, όπου ένα κατεργασμένο μέταλλο ψύχεται ώστε να γίνει πιο εύκαμπτο και λιγότερο εύθραυστο. Έτσι στον αλγόριθμο της Simulated Annealing υπάρχει μία παράμετρος θερμοκρασίας T . Η γενική ιδέα του αλγορίθμου είναι να αποδέχεται λύσεις που δεν βελτιώνουν την αντικειμενική συνάρτηση λαμβάνοντας υπόψη μία πιθανότητα $p(\Delta H, T)$ η οποία εξαρτάται από το κόστος μείωσης $\Delta H = H(y) - H(x)$ και από την τωρινή θερμοκρασία T . Το πιο

συνηθισμένο κριτήριο που επιλέγεται για την μείωση είναι η Metropolis dynamic $p(\Delta H, T) = e^{-\frac{\Delta H}{T}}$. (Deroussi 2016a)(Siarry 2016)



ΣΧΗΜΑ 3.3.4: ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ SIMULATED ANNEALING

Τα πιο συχνά κριτήρια τερματισμού της Simulated Annealing είναι (Talbi 2009) :

- Ο μέγιστος αριθμός επαναλήψεων
- Ο μέγιστος αριθμός επαναλήψεων χωρίς να υπάρξει βελτίωση
- Η μείωση της θερμοκρασίας μέχρι ένα προκαθορισμένο όριο

Παρόλο που τώρα πια η μεθυστική μεθοδολογία Simulated annealing δεν εφαρμόζεται πολύ συχνά σε προβλήματα δρομολόγησης, ήταν από τις πρώτες μεθυστικές που δημοσιεύθηκαν για προβλήματα CVRP. Το 1993 ο Osman παρουσίασε τις λ-interchange moves (Osman 1993). Κατά καιρούς υπήρξαν κάποιες όπως των Lin and Yu σε πρόβλημα κατηγορίας team orientering, (Lin and Yu 2012) (Lin 2013) όπως επίσης και σε πρόβλημα κατηγορίας TTRP with time windows (Lin, Yu, and Lu 2011). Το 2017 από τον Dorling et al (Dorling, Heinrichs, Messier, and Magierowski 2017), οι οποίοι προσέγγισαν το πρόβλημα μεταφοράς αντικειμένων με drone με σκοπό την ελαχιστοποίηση

της κατανάλωσης ενέργειας με περιορισμό τον χρόνο, όπως επίσης και την γενική ελαχιστοποίηση του χρόνου μεταφοράς με περιορισμένη ενέργεια. Εφάρμοσαν και την μέθοδο simulated annealing για να βρουν sub optimal λύσεις σε διάφορα σενάρια

3.3.3 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι γενετικοί αλγόριθμοι παρουσιάστηκαν από τον Holland (Holland 1984). (Luke 2009) Οι γενετικοί αλγόριθμοι μιμούνται τις βασικές αρχές της φύσης όπως αυτές παρουσιάζονται από την θεωρία του Δαρβίνου και του Μέντελ. Βασίζονται σε τρία χαρακτηριστικά (Rothlauf 2011):

- Υπάρχει ένας πληθυσμός από λύσεις.
- Χρησιμοποιούνται τελεστές διασταύρωσης για τον συνδυασμό δύο η περισσότερων λύσεων ώστε να παραχθούν νέες και ίσως καλύτερες λύσεις.
- Υψηλής ποιότητας άτομα επιλέγονται πιο συχνά για αναπαραγωγή κατά την διαδικασία της επιλογής.

Τα βασικά στοιχεία για τον σχεδιασμό γενετικού αλγόριθμου είναι (Burke and Kendall 2013):

- Αρχικοποίηση
- Αξιολόγηση
- Επιλογή
- Διασταύρωση
- Μετάλλαξη
- Αντικατάσταση
- Τερματισμός

ΑΡΧΙΚΟΠΟΙΗΣΗ

Το πρώτο βήμα είναι η αρχικοποίηση του πληθυσμού. Το μέγεθος του πληθυσμού εξαρτάται από τη φύση του προβλήματος. Ο αρχικός πληθυσμός συνήθως δημιουργείται τυχαία. (Burke and Kendall 2013) https://en.wikipedia.org/wiki/Genetic_algorithm

ΑΞΙΟΛΟΓΗΣΗ

Στο βήμα της αξιολόγησης έχουμε την αντικειμενική συνάρτηση. Η αντικειμενική συνάρτηση βαθμολογεί το πόσο κοντά στην λύση του προβλήματος είναι το κάθε στοιχείο του πληθυσμού. (Gendreau and Potvin 2019)

ΕΠΙΛΟΓΗ

Σε αυτό το σημείο γίνεται η επιλογή των γονέων που θα διασταυρωθούν. Η μέθοδος Tournament Selection είναι η πιο διαδεδομένη μέθοδος επιλογής γονέων. Βασίζεται στην σύγκριση της ποιότητας μίας ομάδας από τυχαία επιλεγμένους υποψήφιους γονείς και της επιλογής των καλύτερων από αυτούς. Η διαδικασία της επιλογής γίνεται τόσα τουρνουά όσοι και οι γονείς που θα επιλεγθούν. (Gendreau and Potvin 2019)

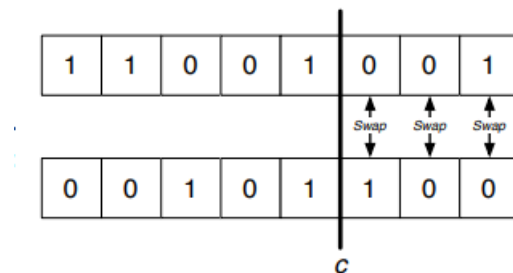
ΔΙΑΣΤΑΥΡΩΣΗ

Οι πιο κλασικοί τρόποι διασταύρωσης σε πίνακες είναι:

- One - Point Crossover.
- Two - Point Crossover.

ONE POINT Crossover

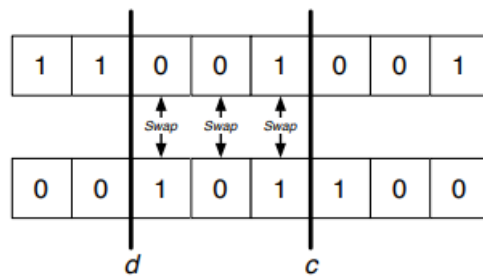
Έχοντας έναν πίνακα μεγέθους L , ο one point crossover επιλέγει έναν αριθμό c από το 1 μέχρι L κλειστό και γίνεται ανταλλαγή των στοιχείων με δείκτη πίνακα μεγαλύτερο από το c . Το πρόβλημα του one point crossover είναι ότι αν έχουμε $c = 1$ ή $c = L$ ουσιαστικά δεν αλλάζει κάτι. (Luke 2009)



ΣΧΗΜΑ 3.3.5: ΠΑΡΑΔΕΙΓΜΑ ONE POINT Crossover

TWO POINT Crossover

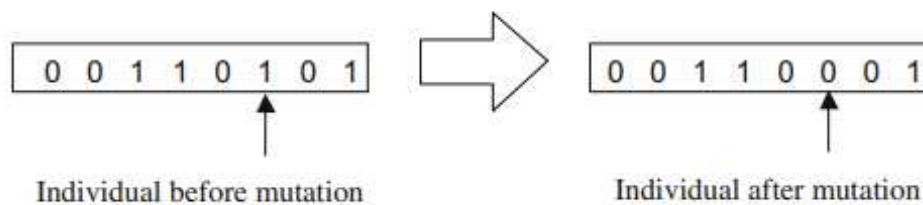
Έχοντας έναν πίνακα μεγέθους L , ο two point crossover επιλέγει δύο αριθμούς c και d και ανταλλάσσουν τα στοιχεία μεταξύ αυτών. (Luke 2009)



ΣΧΗΜΑ 3.3.6: ΠΑΡΑΔΕΙΓΜΑ TWO POINT CROSSOVER

ΜΕΤΑΛΛΑΞΗ

Ένας από τους πιο κλασσικούς τρόπους μετάλλαξης είναι ο bit - flip μετάλλαξη. Στην μετάλλαξη bit flip κάθε ένα bit αντιστρέφεται με μία πιθανότητα p_m γνωστή και ως πιθανότητα μετάλλαξης. (Siarry 2016)



ΣΧΗΜΑ 3.3.7: ΠΑΡΑΔΕΙΓΜΑ BIT - FLIP MUTATION

ΑΝΤΙΚΑΤΑΣΤΑΣΗ

Η πιο διαδεδομένη τεχνική αντικατάστασης είναι η Delete all τεχνική όπου διαγράφονται όλα τα μέλη του τωρινού πληθυσμού και αντικαθίστανται από τα καινούρια που μόλις έχουν δημιουργηθεί. (Burke and Kendall 2013)

ΤΕΡΜΑΤΙΣΜΟΣ

Τα πιο συνηθισμένα κριτήρια τερματισμού γενετικών αλγορίθμων είναι:

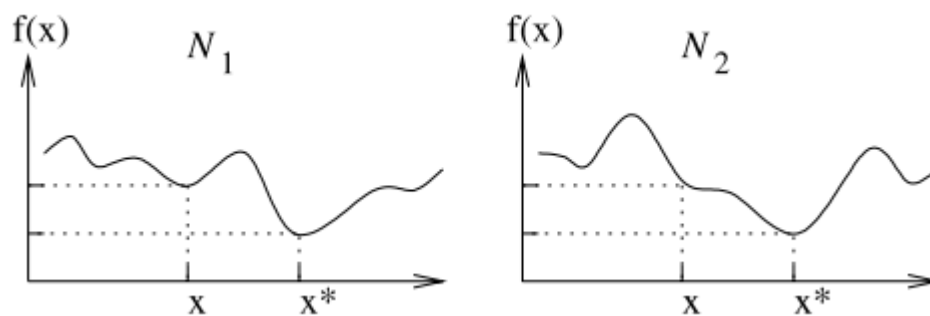
- Μέγιστος αριθμός επαναλήψεων(γενεών).
- Να μην υπάρξει βελτίωση μετά από έναν αριθμό γενεών.
- Να μην υπάρξει καμία αλλαγή σε κανένα μέλος του πληθυσμού.

https://en.wikipedia.org/wiki/Genetic_algorithm

Το 2016 από τον Ferrandez et al (Ferrandez, Harbison, Weber, Sturges, and Rich 2016) όπου με την χρήση του k-means αλγορίθμου για να βρεθούν οι βέλτιστες θέσεις εκκίνησης του drone και την χρήση γενετικού αλγορίθμου για την εύρεση της βέλτιστης διαδρομής του φορτηγού.

3.3.4 VARIABLE NEIGHBORHOOD SEARCH (VNS)

Η αναζήτηση μεταβαλλόμενης γειτονιάς προτάθηκε από τους Mladenovic και Hansen το 1997 (Mladenovic and Hansen 1997). Η βασική της ιδέα είναι η αλλαγή του τρόπου με τον οποίο ορίζεται η γειτονιά μίας λύσης στα πλαίσια μίας τοπικής αναζήτησης. Η VNS βασίζεται σε τρεις παρατηρήσεις: Ένα τοπικό ελάχιστο που βασίζεται σε μία γειτονιά δεν είναι απαραίτητα και τοπικό ελάχιστο σε διαφορετική γειτονιά. Η δομή γειτονιάς του χώρου αναζήτησης εξαρτάται από την μετρική που χρησιμοποιείται και διαφέρει για διαφορετικούς τελεστές αναζήτησης. Ένα ολικό ελάχιστο είναι ολικό ελάχιστο για όλες τις πιθανές γειτονιές. Σε πολλά real - world προβλήματα, τα τοπικά ελάχιστα για μία ή περισσότερες γειτονιές είναι κοντά το ένα με το άλλο. Επίσης τα τοπικά ελάχιστα έχουν ιδιότητες που είναι σχετικές με το ολικό βέλτιστο.



ΣΧΗΜΑ 3.3.8: VNS BASIC

Η βασική ιδέα της VNS όπως φαίνεται στην παραπάνω εικόνα 3.3.8 (Gendreau and Potvin 2019), έχει σαν στόχο την επαναλαμβανόμενη χρήση τοπικής αναζήτησης με χρήση διαφόρων γειτονιών N . (Hansen, Mladenovic, and Pérez 2010) (Rothlauf 2011) Το ολικό βέλτιστο x^* παραμένει το ολικό βέλτιστο λαμβάνοντας υπόψη όλες τις πιθανές γειτονιές. Ωστόσο σε διαφορετικούς γειτονίες έχουμε διαφορετικές γειτονιές, το x μπορεί να είναι τοπικό βέλτιστο λαμβάνοντας υπόψη την γειτονιά N_1 αλλά δεν είναι απαραίτητα και το τοπικό βέλτιστο λαμβάνοντας υπόψη το N_2 . Έτσι κάνοντας τοπική αναζήτηση και ξεκινώντας από το x χρησιμοποιώντας το N_2 μπορούμε να βρούμε το ολικό βέλτιστο. Κατά την αρχικοποίηση ορίζεται ένα σύνολο από k διαφορετικές γειτονιές. (Gendreau and Potvin 2019)

Algorithm 2 Variable Neighborhood Search

```
Select a set of neighborhood structures  $N_k, k \in \{1, \dots, k_{max}\}$ 
Create initial solution  $x$ 
while termination criterion is not met do
   $k = 1$ 
  while  $k < k_{max}$  do
    Shaking: choose a random neighbor  $x' \in N_k(x)$ 
    Local search: perform a local search starting with  $x'$  and return  $x''$  as the local optimum
    with respect to  $N_k$ 
    if  $f(x'') < f(x)$  (minimization problem) then
       $x = x''$ 
       $k = 1$ 
    else
       $k = k + 1$ 
    end if
  end while
end while
```

ΣΧΗΜΑ 3.3.9: VNS ALGORITHM

Η VNS μεθευρητική έχει χρησιμοποιηθεί αποτελεσματικά σε προβλήματα της που ανήκουν στην κατηγορία VRP. (Chen, kuan Huang, and Dong 2010) Επίσης αποτελεσματικά χρησιμοποιείται και σε παραλλαγές του κλασσικού προβλήματος, πιο συγκεκριμένα σε προβλήματα openVRP (Fleszar, Osman, and Hindi 2009), multidepotVRP (Kuo and Wang 2012) και CARP. (Hertz and Mittaz 2001) (Polacek, Doerner, Hartl, and Maniezzo 2008) Μια ακόμα χρήση της VNS σε πρόβλημα όπου drones χρησιμοποιήθηκαν για την φωτογράφιση σημείων που έχουν συμβεί καταστροφές. (Mersheeva and Friedrich 2012)

VND

Σε αυτή την εργασία χρησιμοποιήθηκε και μία παραλλαγή της VNS που ονομάζεται Variable Neighborhood Descent (VND) (Duarte, Mladenović, S'ınchez-Oro Calvo, and Todosijevi 2018).

Η VND χρησιμοποιεί διαδοχικές γειτονίες για την εύρεση βέλτιστου. Αρχικά θέτουμε ένα σύνολο από γειτονίες K_{max} . Οπου k_1 η πρώτη γειτονιά που χρησιμοποιείται έχοντας σαν x την αρχική λύση. Αν δεν υπάρξει βελτίωση της αρχικής λύσης με την χρήση της πρώτης γειτονιάς τότε χρησιμοποιούμε την $k_1 + 1$. Αν υπάρξει βελτίωση την τωρινής βέλτιστης λύσης, επιστρέφουμε στην πρώτη γειτονιά και ξαναξεκινάει η αναζήτηση. Αυτή η τεχνική φαίνεται να είναι αποτελεσματική στην περίπτωση που το τοπικό ελάχιστο της γειτονιάς k_i να μην είναι τοπικό ελάχιστο της γειτονιάς k_j . (Gendreau and Potvin 2019)

```

Function VND ( $x, k_{max}$ )
1  $k \leftarrow 1$ 
2 repeat
3    $x' \leftarrow \arg \min_{y \in N_k(x)} f(y)$  // Find the best neighbor in  $N_k(x)$ 
4    $x, k \leftarrow \text{NeighborhoodChange}(x, x', k)$  // Change neighborhood
   until  $k = k_{max}$ 
return  $x$ 

```

ΣΧΗΜΑ 3.3.10: VND ALGORITHM.

Ο σχεδιασμός του αλγόριθμου VND σχετίζεται κυρίως με την επιλογή των γειτονιών και την σειρά εφαρμογής τους. Η πολυπλοκότητα των γειτονιών πρέπει να ληφθεί υπόψη. Όσο πιο μεγάλες είναι οι γειτονιές τόσο πιο πολύ χρόνο χρειάζεται ο αλγόριθμος VND. Μία δημοφιλής στρατηγική ταξινόμησης των γειτονιών είναι όσο αυξάνεται η πολυπλοκότητα.

RVNS

Μία ακόμα παραλλαγή της VNS είναι η Reduced VNS που όμως δεν υλοποιείται στην παρούσα εργασία (Mladenović 2003). Σε αυτή την μέθοδο βρίσκουμε ένα τυχαίο σημείο το συγκρίνουμε με το ήδη υπάρχον και έχουμε ενημέρωση σε περίπτωση βελτίωσης. Υποθέτουμε επίσης ότι έχει επιλεγθεί μία συνθήκη τερματισμού, όπως μέγιστη cpu time ή ο μέγιστος αριθμός από επαναλήψεις πριν να έχουμε δύο βελτιώσεις. (Gendreau and Potvin 2019)

```

Function RVNS( $x, k_{max}, t_{max}$ )
1 repeat
2    $k \leftarrow 1$ 
3   repeat
4      $x' \leftarrow \text{Shake}(x, k)$ 
5      $x, k \leftarrow \text{NeighborhoodChange}(x, x', k)$ 
     until  $k = k_{max}$ 
6    $t \leftarrow \text{CpuTime}()$ 
until  $t > t_{max}$ 
return  $x$ 

```

ΣΧΗΜΑ 3.3.11: REDUCED VNS ALGORITHM

Η συνάρτηση shake δημιουργεί ένα τυχαίο σημείο x' χρησιμοποιώντας την k οστη γειτονι-
 ά.(Gendreau and Potvin 2019)

```

Function Shake( $x, k$ )
1  $w \leftarrow \lfloor 1 + \text{Rand}(0, 1) \times |N_k(x)| \rfloor$ 
2  $x' \leftarrow x^w$ 
return  $x'$ 

```

ΣΧΗΜΑ 3.3.12: SHAKE VNS ALGORITHM

Η RVNS είναι πολύ χρήσιμη σε προβλήματα μεγάλης κλίμακας όπου η τοπική αναζήτηση έχει μεγάλο κόστος. Μπορεί επίσης να χρησιμοποιηθεί για την εύρεση αρχικών λύσεων σε μεγάλα προβλήματα. Έχει παρατηρηθεί ότι η καλύτερη τιμή για την παράμετρο K_{max} είναι συχνά μεταξύ 2 και 3. Επιπροσθέτως σαν κριτήριο τερματισμού θέτεται συχνά ένας μέγιστος αριθμός μεταξύ δύο βελτιώσεων.

ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ VNS

SKEWED VNS (SVNS)

Η Skewed VNS (SVNS) μέθοδος αντιμετωπίζει το πρόβλημα της εξερεύνησης κοιλάδων μακριά από την υφιστάμενη λύση. (Bouhmala, Hjelmervik, and Øvergaard 2015) Πραγματικά μόλις βρεθεί η καλύτερη λύση σε μία μεγάλη περιοχή πρέπει να γίνει μεγάλη εξερεύνηση για να έχουμε μία βελτίωση. Λύσεις που βρίσκονται τυχαία σε μακρινές περιοχές και η λύση μπορεί να διαφέρει σημαντικά από την υφιστάμενη λύση και η VNS μπορεί να φτάσει στο σημείο να μοιάζει με Multistart ευρετική, όπου δημιουργούνται τυχαίες λύσεις και αυτό έχει αποδειχθεί μη αποτελεσματικό. Έτσι για να αποζημιωθούμε από την απόσταση που διανύθηκε έχουμε την Skewed VNS (Hansen and Mladenović 2001).

VARIABLE NEIGHBORHOOD DECOMPOSITION SEARCH (VNDS)

Variable Neighborhood Decomposition search (VNDS) επεκτείνει την βασική VNS σε ένα σχήμα VNS δύο επιπέδων που βασίζεται στην διάσπαση του προβλήματος (Hansen, Mladenović, and Perez-Britos 2001).

ΚΕΦΑΛΑΙΟ 4

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ ΛΥΣΗΣ

4.1 ΕΙΣΑΓΩΓΗ

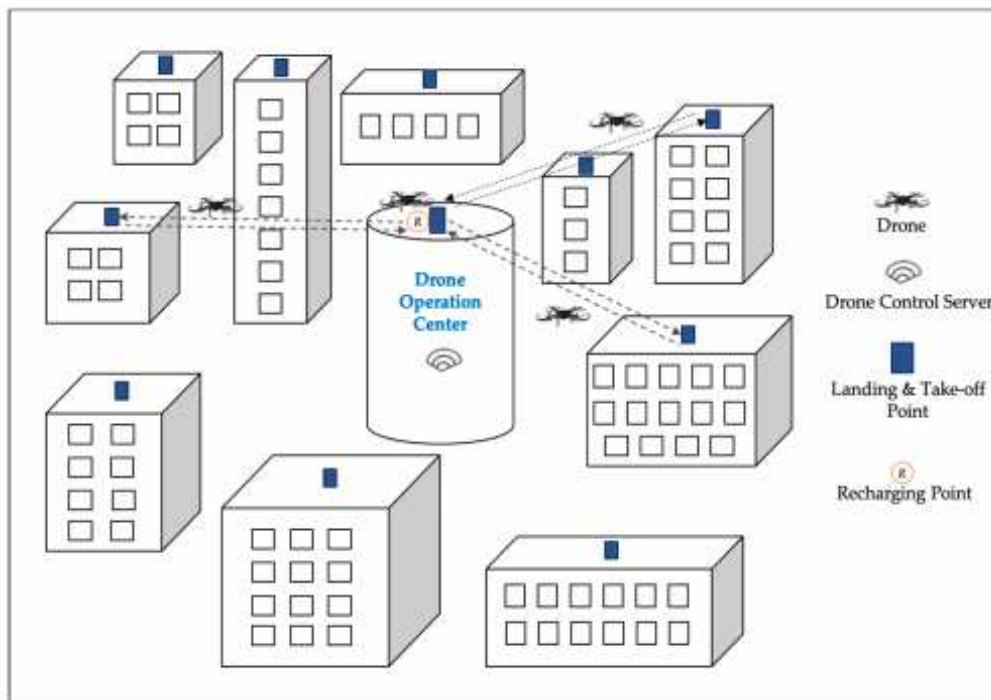
Στο παρόν κεφάλαιο αρχικά γίνεται η περιγραφή του προβλήματος που θα επιλυθεί. Έπειτα περιγράφονται οι τρόποι εύρεσης αρχικής λύσης και λύσης βελτίωσης. Το κεφάλαιο χωρίζεται σε 3 ενότητες. Στην πρώτη ενότητα παρουσιάζεται το πρόβλημα. Στην δεύτερη ενότητα παρουσιάζονται οι αλγόριθμοι δημιουργίας αρχικής λύσης και στην τρίτη και τελευταία ενότητα παρουσιάζονται οι αλγόριθμοι βελτίωσης της αρχικής λύσης.

4.2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Το πρόβλημα μας είναι η δημιουργία ενός προγράμματος μεταφορών αντικειμένων με την χρήση drones επωφελούμενοι τις οροφές κτηρίων.(Kim, Moon, and Jung 2020) Για την λύση του προβλήματος γίνονται οι παρακάτω υποθέσεις:

- Το κάθε ένα από τα κτήρια έχει οροφή.
- Η οροφή ενός εκ των κτηρίων χρησιμοποιείται σαν drone operation center.
- Τα drones λειτουργούν σε hub and spoke δίκτυο, το κάθε ένα από τα drones μπορεί να κουβαλήσει το πολύ ένα πακέτο. Επίσης κινούνται μόνο από το drone operation center σε κάποιο από τα κτήρια και πίσω πάλι στο operation center (Ferrandez, Harbison, Weber, Sturges, and Rich 2016).
- Χρησιμοποιούνται περισσότερα από ένα drones. Ο αριθμός τους είναι προκαθορισμένος.
- Τα drones χρησιμοποιούνται για να υποστηρίξουν τους κλασσικούς τρόπους μεταφοράς και όχι να τους αντικαταστήσουν.
- Ο αριθμός της ζήτησης από κάθε κτήριο είναι γνωστός.

- Η απόσταση του κεντρικού κτηρίου από τα υπόλοιπα κτήρια μπορεί να υπολογιστεί με χρήση της ευκλείδειας απόστασης.



ΣΧΗΜΑ 4.2.1: DRONE BASED PARCEL DELIVERY USING THE ROOFTOPS OF THE BUILDINGS

Επιπροσθέτως με τις πιο πάνω υποθέσεις, το βάρος των φορτίων περιορίζεται περίπου στα 2 κιλά, γι' αυτό τον λόγο τα εμπορικά drones δεν μπορούν να μεταφέρουν περισσότερα από ένα αντικείμενο. Τα drones μπορούν να μετακινηθούν μέχρι περίπου 10 - 15 χιλιόμετρα με καλές καιρικές συνθήκες. Μπορούμε να φορτίσουμε την μπαταρία του drone οποιαδήποτε στιγμή θέλουμε, αρκεί η μπαταρία να μην είναι ήδη φορτισμένη. Οι γενικές περιγραφές των drones συνοψίζονται στην παρακάτω εικόνα 4.2.2. (Kim, Moon, and Jung 2020)

Τα drones εκτελούν μεταφορές για 9 ώρες. Η απογείωση, η προσγείωση, η φόρτωση και η εκφόρτωση των πακέτων θεωρούνται σαν επιπλέον δουλειές. Υποθέτουμε ότι αυτές οι δουλειές χρειάζονται 15 λεπτά, υποθέτουμε επίσης ότι τα drones ξεκινάνε με εντελώς φορτισμένη την μπαταρία τους. Τέλος δεν λαμβάνουμε υπόψη μας τις καιρικές συνθήκες.

Table 1. Delivery drone specifications in the case study.

Item	Unit	Value
Vehicle mass	kg	8
(Maximum) Payload mass	kg	2
(Average) Cruising velocity	km/h	50
Power transfer efficiency for motor and propeller	-	0.5
Lift to drag ratio	-	3
Power consumption of electronics	kWh	0.1
Battery consumption	kWh	0.6
Battery capacity	kWh	1.54
Recharging time (to full)	hour	1.5
Operating hours	hour	9
Incidental working hours (take-off, landing, loading and dropping packages)	hour	0.25

ΣΧΗΜΑ 4.2.2: EXAMPLE DRONE SPECIFICATIONS

Η χειρότερη περίπτωση κατανάλωσης μπαταρίας υπολογίζεται από την εξίσωση που προτάθηκε από τον D'Andrea (D'Andrea 2014). Μιας και η κατανάλωση μπαταρίας είναι ύψιστης σημασίας για την ασφάλεια πτήσεων των drones, γι' αυτό τον λόγο χρησιμοποιείται η χειρότερη περίπτωση για να αποφευχθούν τα προβλήματα κατά την διάρκεια της πτήσης.

4.3 ΜΕΘΟΔΟΙ ΑΡΧΙΚΟΠΟΙΗΣΗΣ

Για την εύρεση αρχικής λύσης δοκιμάστηκαν 3 αλγόριθμοι. Στις παρακάτω ενότητες παρουσιάζονται ο ψευδοκώδικας και ένα παράδειγμα για τον κάθε αλγόριθμο.

4.3.1 ΠΡΩΤΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ

Η κεντρική ιδέα του αλγορίθμου είναι το μέγεθος των blocks. Αρχικά γίνεται ταξινόμηση των blocks από το μικρότερο στο μεγαλύτερο. Εισάγουμε το πρώτο block στο drone με τον περισσότερο ελεύθερο χρόνο. Ο αλγόριθμος αναφέρεται στο άρθρο (Kim, Moon, and Jung 2020).

Algorithm 1 construction heuristic 1

Require: $G = \text{drones}, \text{listofblocks}, \text{nofdrones}, \text{chargingtimetofull}$

procedure CONSTRUCTION 1(G, T)

2: initialize closedrones = []

4: **while** len(drones) > 0 and len(listofblocks) > 0 and len(closedrones) != nofdrones **do**
 Stack the smallest block from the listofblocks to an open container which has the largest space

6: Create a power set made out of delivery blocks in the container, and select a subset whose summed battery consumption is most similar to the battery amount
 Move blocks in the select a subset to the end of the container. Remove the charging

8: Calculate the total length of the container
 if total length > operation time **then**

10: Withdraw lines [5,6,7,8]
 Close the container

12: **else**
 Remove the first block from listofblocks

14: **end if**
 end while

16: **end procedure**

4.3.2 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για την κατασκευή αρχικής λύσης με τον Αλγόριθμο 1, αρχικά εισαγάγουμε όσα περισσότερα πακέτα μπορούμε από το μικρότερο (αυτό με λιγότερο συνολικό χρόνο μεταφοράς και φόρτισης) στο μεγαλύτερο, με την προϋπόθεση ο συνολικός χρόνος μεταφοράς να μην ξεπεράσει της 9 ώρες. Μετά από κάθε εισαγωγή πακέτου φτιάχνουμε ένα δυναμοσύνολο με πακέτα στα οποία αθροίζουμε τον χρόνο φόρτισης και διαλέγουμε το πιο κοντινό στον χρόνο συνολικής φόρτισης ενός drone. Ο λόγος για τον οποίο γίνεται αυτό είναι επειδή μετά το πέρας των 9 ωρών λειτουργίας δεν χρειάζεται να ξαναφορτίσουμε τα drones. Αν κάποιο πακέτο όταν εισαχθεί ο συνολικός χρόνος μεταφοράς για το συγκεκριμένο drone ξεπεράσει τις 9 ώρες το πακέτο εξάγεται και το container αυτού του drone κλείνει και δεν είναι πια διαθέσιμο. Για περισσότερες πληροφορίες ανατρέξτε στον Αλγόριθμο 1. Σύμφωνα με τα παραπάνω και έστω ότι έχουμε να διαμοιράσουμε 33 πακέτα με την χρήση δυο drones.

Name of the block	Total Length
0	0.688
1	0.756
2	0.806
3	0.828
4	0.842
5	0.914
6	0.932
7	0.95
8	1.121
9	1.14

ΠΙΝΑΚΑΣ 4.1: ΣΥΝΟΛΙΚΟ ΜΕΓΕΘΟΣ ΤΩΝ ΠΑΚΕΤΩΝ (TOTAL LENGTH OF BLOCKS)

NAME OF DRONE	REMAINING SPACE	CUSTOMER
0	0.04	0 0 2 2 2 3 3 4 5 1 1 6
1	0.74	0 0 0 2 2 3 3 3 5 1 1 6 6
	(REMAINING BLOCKS)	6 7 7 8 8 8 9 9

ΠΙΝΑΚΑΣ 4.2: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1

Στον πίνακα 4.1 βλέπουμε την σειρά με την οποία εισήχθησαν τα πακέτα. Όπως φαίνεται ο συνολικός χρόνος των πακέτων (φόρτισης και μεταφοράς) ξεκινάει από το μικρότερο και καταλήγει στο μεγαλύτερο. Δηλαδή η σειρά εισαγωγής των πακέτων είναι: 0,1,2,3,4,5,6,7,8,9. Από τον πίνακα 4.2 προκύπτει ότι με την χρήση του αλγόριθμου construction 1 στο container 0 του drone 0 πριν αυτό κλείσει εισάγονται 12 από τα 33 συνολικά πακέτα. Η σειρά εισαγωγής έχει ως εξής: 0,0,2,2,2,3,3,4,5,1,1,6. Με την εισαγωγή αυτών των πακέτων ο ελεύθερος χώρος που απομένει για το container του drone 0 είναι 0.040. Επίσης από τον πίνακα ;; μπορούμε να εξάγουμε πληροφορίες

και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container 1 του drone 1. Στο container 1 του drone 1 εισήχθησαν πριν αυτό κλείσει 13 από τα 33 συνολικά πακέτα. Η σειρά εισαγωγής είναι η εξής: 0,0,0,2,2,3,3,3,5,1,1,6,6. Με την εισαγωγή αυτών των πακέτων στο container του drone 1 ο ελεύθερος χώρος που απομένει είναι 0.740. Τέλος από τον πίνακα 4.2 βλέπουμε ότι με την χρήση του αλγόριθμου construction 1 υπάρχουν 8 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα δύο container. Τα πακέτα αυτά είναι: 6,7,7,8,8,8,9,9.

4.3.3 ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ

Στην παρούσα ενότητα παρουσιάζεται μία μέθοδος αρχικοποίησης που δημιουργήσα, με σκοπό την εύρεση καλύτερης αρχικής λύσης συγκριτικά με την 4.2. Η κεντρική ιδέα του αλγόριθμου είναι ο χρόνος φόρτισης. Αρχικά γίνεται ταξινόμηση των blocks από το μεγαλύτερο charging time στο μικρότερο. Εισάγουμε το πρώτο block στο drone με τον περισσότερο ελεύθερο χρόνο.

Algorithm 2 construction heuristic 2

Require: $G = \text{drones}, \text{listofblocks}, \text{nofdrones}, \text{chargingtimetofull}$

```

procedure CONSTRUCTION 2( $G, T$ )
2:   initialize closedrones = []

4:   while len(drones) > 0 and len(listofblocks) > 0 and len(closedrones) != nofdrones do
       sort list of blocks from the largest charging time
6:     find the container with the most free space
       Stack the first block from the listofblocks
8:     Create a power set made out of delivery blocks in the container, and select a subset whose summed
       battery consumption is most similar to the battery amount
       Move blocks in the select a subset to the end of the container. Remove the charging
10:    Calculate the total length of the container
       if total length > operation time then
12:      Withdraw lines [7,8,9,10]
       Close the container
14:    else
       Remove the first block from listofblocks
16:    end if
       end while
18: end procedure

```

4.3.4 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για την κατασκευή αρχικής λύσης με τον Αλγόριθμο 2, αρχικά εισαγάγουμε όσα περισσότερα πακέτα μπορούμε από το μεγαλύτερο (αυτό με περισσότερο συνολικό χρόνο φόρτισης) στο μικρότερο, με την προϋπόθεση ο συνολικός χρόνος μεταφοράς να μην ξεπεράσει της 9 ώρες. Μετά από κάθε εισαγωγή πακέτου φτιάχνουμε ένα δυναμοσύνολο με πακέτα στα οποία αθροίζουμε τον χρόνο φόρτισης και διαλέγουμε το πιο κοντινό στον χρόνο συνολικής φόρτισης ενός drone. Ο λόγος για τον οποίο γίνεται αυτό είναι επειδή μετά το πέρας των 9 ωρών λειτουργίας δεν χρειάζεται να ξαναφορτίσουμε τα drones. Αν κάποιο πακέτο όταν εισαχθεί ο συνολικός χρόνος μεταφοράς για

Name of the block	Charging time
9	0.65
8	0.64
7	0.54
6	0.53
5	0.52
4	0.48
3	0.47
2	0.46
1	0.43
0	0.39

ΠΙΝΑΚΑΣ 4.3: ΧΡΟΝΟΣ ΦΟΡΤΙΣΗΣ ΓΙΑ ΤΗΝ ΜΕΤΑΦΟΡΑ ΠΑΚΕΤΩΝ (*CHARGING TIME OF DRONES*)

NAME OF DRONE	REMAINING SPACE	CUSTOMER										
0	0.179	6	6	5	9	8	2	7	7	3	3	3
1	0.147	5	9	8	8	2	2	6	6	4	3	3
	<i>(REMAINING BLOCKS)</i>	2	2	1	1	1	1	0	0	0	0	0

ΠΙΝΑΚΑΣ 4.4: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 2

το συγκεκριμένο drone ξεπεράσει τις 9 ώρες το πακέτο εξάγεται και το container αυτού του drone κλείνει και δεν είναι πια διαθέσιμο. Για περισσότερες πληροφορίες ανατρέξτε στον Αλγόριθμο construction 2. Σύμφωνα με τα παραπάνω και έστω ότι έχουμε να διαμοιράσουμε 33 πακέτα με την χρήση δύο drones.

Στον πίνακα 4.3 βλέπουμε την σειρά με την οποία εισήχθησαν τα πακέτα. Όπως φαίνεται ο χρόνος φόρτισης που απαιτείται για την παράδοση των πακέτων ξεκινάει από το μεγαλύτερο και καταλήγει στο μικρότερο.

Από τον πίνακα 4.4 προκύπτει ότι με την χρήση του αλγόριθμου construction 2 στο container του πρώτου drone πριν αυτό κλείσει εισάγονται 11 από τα 33 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων ο ελεύθερος χώρος που απομένει για το container 1 του drone 1 είναι 0.179. Επίσης από τον πίνακα 4.4 μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 2 στο container 0 του drone 0. Στο container 0 του drone 0 εισήχθησαν πριν αυτό κλείσει 11 από τα 33 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container 0 του drone 0 ο ελεύθερος χώρος που απομένει είναι 0.147. Τέλος από τον πίνακα 4.4 βλέπουμε ότι με την χρήση του αλγόριθμου construction 2 υπάρχουν 11 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα δύο container.

4.3.5 ΤΡΙΤΗ ΜΕΘΟΔΟΣ ΑΡΧΙΚΟΠΟΙΗΣΗΣ

Στην παρούσα ενότητα παρουσιάζεται μία ακόμη μέθοδος αρχικοποίησης που δημιουργήσα, με σκοπό την εύρεση καλύτερης αρχικής λύσης συγκριτικά με την 4.2 και την 4.4. Ο αλγόριθμος 3 βασίζεται στο όνομα του container. Τα ονόματα των containers είναι αριθμοί από το 0-9, ανάλογα τον τύπο του προβλήματος που λύνεται. Αν το όνομα του container διαιρείται ακριβώς με το 2 τότε παίρνει το πρώτο block από την λίστα με τα blocks, αλλιώς παίρνει το τελευταίο block από την λίστα με τα blocks.

Algorithm 3 construction heuristic 3

Require: $G = \text{drones}, \text{listofblocks}, \text{nofdrones}, \text{chargingtimetofull}$

```
procedure CONSTRUCTION 3( $G, T$ )
2:   initialize closedrones = []

4:   while len(drones) > 0 and len(listofblocks) > 0 and len(closedrones) != nofdrones do
       for  $i \leftarrow 0$  to drones do
6:         if len(listofblocks) == 0) then
           Break
8:         else
           Pass
10:        end if
        if ( $i.\text{getName}()/2 == 0$ ) then
12:          Add the first block of the listofblocks to the i container
        else
14:          Add the last block of the listofelements to the i container
        end if
16:        Create a power set made out of delivery blocks in the container, and select a subset whose summed battery consumption is most similar to the battery amount
        Move blocks in the select a subset to the end of the container. Remove the charging
18:        Calculate the total length of the container
        if total length > operation time then
20:          Withdraw lines [17,18,16,] [14]or[12]
          Close the container
22:        else
          Remove the first block from listofblocks
24:        end if
       end for
26:   end while
end procedure
```

4.3.6 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για την κατασκευή αρχικής λύσης με τον Αλγόριθμο 3, αρχικά εισαγάγουμε όσα περισσότερα πακέτα μπορούμε ανάλογα με το όνομα του container δηλαδή αν το όνομα είναι άρτιος αριθμός εισαγάγουμε το μικρότερο, αν είναι περιττός το μεγαλύτερο, με την προϋπόθεση ο συνολικός χρόνος μεταφοράς να μην ξεπεράσει τις 9 ώρες. Μετά από κάθε εισαγωγή πακέτου φτιάχνουμε ένα δυναμικό με πακέτα στα οποία αθροίζουμε τον χρόνο φόρτισης και διαλέγουμε το πιο κοντινό στον χρόνο συνολικής φόρτισης ενός drone. Ο λόγος για τον οποίο γίνεται αυτό είναι επειδή μετά το

NAME OF DRONE	REMAINING SPACE	CUSTOMER											
0	0.889	8	8	8	7	7	6	6	9	9			
1	0.691	0	0	0	0	0	1	1	1	1	2	2	2
	(REMAINING BLOCKS)	2	3	3	3	3	3	4	5	5	6	6	

ΠΙΝΑΚΑΣ 4.5: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 3

πέρασ των 9 ωρών λειτουργίας δεν χρειάζεται να ξαναφορτίσουμε τα drones. Αν κάποιο πακέτο όταν εισαχθεί ο συνολικός χρόνος μεταφοράς για το συγκεκριμένο drone ξεπεράσει τις 9 ώρες, το πακέτο εξάγεται και το container αυτού του drone κλείνει και δεν είναι πια διαθέσιμο. Για περισσότερες πληροφορίες ανατρέξτε στον Αλγόριθμο construction 3. Σύμφωνα με τα παραπάνω και έστω ότι έχουμε να διαμοιράσουμε 33 πακέτα με την χρήση δύο drones.

Από τον πίνακα 4.5 προκύπτει ότι με την χρήση του αλγόριθμου construction 3 στο container 1 του drone 1 πριν αυτό κλείσει εισάγονται 9 από τα 33 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων ο ελεύθερος χώρος που απομένει είναι για το container του drone 1 είναι 0.889. Επίσης από τον πίνακα 4.5 μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 3 στο container 0 του drone 0. Στο container 0 του drone 0 εισήχθησαν πριν αυτό κλείσει 13 από τα 33 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 0 ο ελεύθερος χώρος που απομένει είναι 0.691. Τέλος από τον πίνακα 4.5 βλέπουμε ότι με την χρήση του αλγόριθμου construction 3 υπάρχουν 11 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα δύο container.

Η σειρά εισαγωγής δεν μπορεί να αποτυπωθεί σε πίνακα λόγω της φύσης του αλγόριθμου.

4.4 ΜΕΘΟΔΟΙ ΒΕΛΤΙΩΣΗΣ

Για να βελτιωθεί η αρχική λύση δοκιμάστηκαν 3 διαφορετικοί αλγόριθμοι. Στις παρακάτω ενότητες παρουσιάζονται ο ψευδοκώδικας και ένα παράδειγμα για τον κάθε αλγόριθμο.

4.4.1 ΠΡΩΤΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ

Η κεντρική ιδέα του αλγορίθμου είναι η ανταλλαγή πακέτων μεταξύ των containers για την απελευθέρωση χώρου και για την εισαγωγή πακέτου που περισσεύει. Ο αλγόριθμος αναφέρεται στο άρθρο (Kim, Moon, and Jung 2020).

Algorithm 4 Improvement heuristic 1

Require: $G = \text{containers, remainingblocks, nofdrones, chargingtimetofull}$

procedure IMPROVEMENT 1(G, T)

2: **while** the total remaining space of containers \geq the size of the new blocks to stuck **do**

4: Sort containers in descending order of the total remaining space of the container
 Select a container that has the largest remaining space (1st) and a container that has the second-largest remaining space(2nd)

6: Make combinations that have 2 elements made out of the intersection between the destinations of two selected containers
 select combination which has the largest gap between 2 elements on condition that this gap does not exceed the remaining space of the 2nd container.

8: Remove a smaller destination block of the selected combination from the 2nd container and stack a larger destination block of the selected combination to the 2nd container.
 With the second container

10: Create a power set made out of delivery blocks in the container, and select a subset whose summed battery consumption is most similar to the battery amount
 Move blocks in the select a subset to the end of the container.Remove the charging

12: Calculate the total length of the container
 if total length $>$ operation time **then**

14: Withdraw lines [13,12,11,10]
 Return to line 7 and select combination which has the next larger gap

16: **else**
 Remove a larger destination block of the selected combination from the 1st container and stack a smaller destination block of the selected combination to the 1st container

18: **end if**
 if there is no appropriate combination between the 1st and the 2nd containers **then**

20: Return to line and select the 1st container and a container that has the next largest space(3,4,5..)
 else

22: Pass

end if

24: **if** there is enough space to stack new block **then**
 Stack the new block
 Implement [10,11,12]

26: **if** total length $>$ operation time **then**

28: Withdraw lines [27,26]

else

30: Pass

end if

32: **else**
 Pass

34: **end if**
 if there is no appropriate combination among all containers **then**

36: Break

else
 Pass

38: **end if**

40: **end while**
end procedure

4.4.2 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για την βελτίωση του προβλήματος γίνονται ανταλλαγές πακέτων από το container με τον περισσότερο ελεύθερο χώρο σε κάποιο από τα υπόλοιπα. Το container με τον περισσότερο χώρο δίνει ένα μεγαλύτερο πακέτο ενώ το άλλο container δίνει ένα μικρότερο πακέτο. Έτσι απελευθερώνεται περισσότερος χώρος στο μεγαλύτερο container και έτσι μπορεί να είναι δυνατή η εισαγωγή καινούργιου πακέτου από αυτά που περισσεύουν.

Επειδή το μικρότερο block που περισσεύει δεν είναι μικρότερο ή ίσο σε μέγεθος με τον συνολικό ελεύθερο χώρο των containers και επειδή αυτό είναι η προϋπόθεση προσπάθειας βελτίωσης του προβλήματος το συγκεκριμένο πρόβλημα δεν μπορεί να βελτιωθεί με τον συγκεκριμένο αλγόριθμο. Για περισσότερες πληροφορίες ανατρέξτε στον Αλγόριθμο Improvement 1. Έτσι για την ανάγκη παρουσίασης χρησιμοποιείται ένα διαφορετικό πρόβλημα όπου έχει συνολικά 24 πακέτα που πρέπει να μεταφερθούν.

NAME OF DRONE	REMAINING SPACE	CUSTOMER
0	1.373	3 3 6 6 7 7 4 5 1 0
1	0.884	5 3 3 6 6 4 2 7 7 1
	(REMAINING BLOCKS)	8 8 9 9

ΠΙΝΑΚΑΣ 4.6: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1

NAME OF DRONE	REMAINING SPACE	CUSTOMERS
0	0.312	3 3 6 6 7 7 8 1 0 5 5
1	0.317	3 3 6 6 4 2 4 7 7 1
	(REMAINING BLOCKS)	8 9 9

ΠΙΝΑΚΑΣ 4.7: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 1

Από τον πίνακα 4.6 προκύπτει ότι με την χρήση του αλγόριθμου construction 1 στο container του πρώτου drone πριν αυτό κλείσει εισάγονται 10 από τα 24 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων υπάρχει διαθέσιμος χώρος για το container του drone 1 και συγκεκριμένα 1.373. Επίσης από τον πίνακα 4.6 μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container του δεύτερου drone. Στο container του δεύτερου drone εισήχθησαν πριν αυτό κλείσει 10 από τα 24 συνολικά πακέτα. Η σειρά εισαγωγής είναι η εξής: 5,3,3,6,6,4,2,7,7,1. Με την εισαγωγή αυτών των πακέτων στο container του drone 2 ο ελεύθερος χώρος που απομένει είναι 0.884. Τέλος από τον πίνακα 4.6 βλέπουμε ότι με την χρήση του αλγόριθμου construction 1 υπάρχουν 4 πακέτα τα οποία περισσεύουν και δεν έχουν εισαχθεί σε κανένα από τα δύο container. Από τον πίνακα 4.7 διαπιστώνουμε ότι έχει γίνει βελτίωση του προβλήματος αφού έχει εισαχθεί καινούριο πακέτο με μεταβολές στον υπολειπόμενο χώρο.

4.4.3 ΔΕΥΤΕΡΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ

Στην παρούσα ενότητα παρουσιάζεται μια μέθοδος βελτίωσης την οποία δημιούργησα με σκοπό την μεγαλύτερη βελτίωση της αρχικής λύσης συγκριτικά με την πρώτη μέθοδο βελτίωσης. Από το container με τον περισσότερο ελεύθερο χώρο παίρνουμε το πρώτο block. Για κάθε ένα από τα υπόλοιπα containers ελέγχουμε αν υπάρχει ελεύθερος χώρος για την εισαγωγή του block που διαγράφηκε από το πρώτο. Αν δεν υπάρχει ελεύθερος χώρος και δεν μπορούμε να εισάγουμε το block, τότε επιστρέφουμε το block στο αρχικό container και τερματίζεται ο αλγόριθμος. Αν το block εισαχθεί σε κάποιο από τα containers τότε γίνεται προσπάθεια εισαγωγής ενός από τα πακέτα που έχουν περισσέψει από την αρχική λύση. Αν δεν είναι δυνατό να γίνει η εισαγωγή του block τότε υπολογίζεται ο ελεύθερος χώρος που απομένει στα containers και γίνεται πάλι η ίδια διαδικασία.

Algorithm 5 Improvement heuristic 2

Require: $G = \text{containers}, \text{remainingblocks}, \text{nofdrones}, \text{chargingtimetofull}$

procedure IMPROVEMENT 2(G, T)

```
2:
    while number of remaining blocks > 0 and the total remaining space of containers  $\geq$  the size of the new
    blocks to stack do
4:     Sort the containers by remaining space in descending order
       remove the first block of the 1st container
6:     if the remaining space of the 2nd container is larger than the removed block then
       Stack the removed block
8:     Create a power set made out of delivery blocks in the container, and select a subset whose summed
       battery consumption is most similar to the battery amount
       Move blocks in the select a subset to the end of the container. Remove the charging
10:    Calculate the total length of the container
       if total length > operation time then
12:        Withdraw lines [10,9,8,7,5]
           Break and terminate
14:    else
       Stack to the 1st container a block from the remaining blocks
16:    Create a power set made out of delivery blocks in the container, and select a subset whose
       summed battery consumption is most similar to the battery amount
       Move blocks in the select a subset to the end of the container. Remove the charging
18:    Calculate the total length of the container
       if total length > operation time then
20:        Withdraw lines [17,16,15,14]
           Break
22:    else
       Calculate the total remaining space of all containers
24:    Break
       end if
26:    end if
       else
28:        return the removed block to the 1st container
           Break
30:    end if
       end while
32: end procedure
```

4.4.4 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για την βελτίωση του προβλήματος γίνεται μεταφορά πακέτου από το container με τον περισσότερο ελεύθερο χώρο σε κάποιο από τα υπόλοιπα. Το container με τον περισσότερο χώρο δίνει το μικρότερο πακέτο που έχει σε ένα από τα containers που χωράει. Έτσι απελευθερώνεται περισσότερος χώρος στο μεγαλύτερο container και έτσι μπορεί να είναι δυνατή η εισαγωγή καινούριου πακέτου από αυτά που περισσεύουν.

Επειδή το μικρότερο block που περισσεύει δεν είναι μικρότερο ή ίσο σε μέγεθος με τον συνολικό ελεύθερο χώρο των container και επειδή αυτό είναι η προϋπόθεση προσπάθειας βελτίωσης του προβλήματος το συγκεκριμένο πρόβλημα δεν μπορεί να βελτιωθεί με τον συγκεκριμένο αλγόριθμο. Έτσι για την ανάγκη παρουσίας χρησιμοποιείται ένα διαφορετικό πρόβλημα όπου έχει συνολικά 24 πακέτα που πρέπει να μεταφερθούν.

NAME OF DRONE	REMAINING SPACE	CUSTOMER
0	1.373	3 3 6 6 7 7 4 5 1 0
1	0.884	5 3 3 6 6 4 2 7 7 1
	(REMAINING BLOCKS)	8 8 9 9

ΠΙΝΑΚΑΣ 4.8: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1

NAME OF DRONE	REMAINING SPACE	CUSTOMERS
0	0.709	3 6 6 7 7 4 8 5 1 0
1	0.05	5 3 3 6 6 4 2 3 7 7 1
	(REMAINING BLOCKS)	8 9 9

ΠΙΝΑΚΑΣ 4.9: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 2

Από τον πίνακα 4.8 προκύπτει ότι με την χρήση του αλγόριθμου construction 1 στο container του πρώτου drone πριν αυτό κλείσει εισάγονται 10 από τα 24 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων υπάρχει διαθέσιμος χώρος για το container του drone 1 και συγκεκριμένα 1.373. Επίσης από τον πίνακα 4.8 μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container του δεύτερου drone. Στο container του δεύτερου drone εισήχθησαν πριν αυτό κλείσει 10 από τα 24 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 2 ο ελεύθερος χώρος που απομένει είναι 0.884. Τέλος από τον πίνακα 4.8 βλέπουμε ότι με την χρήση του αλγόριθμου construction 1 υπάρχουν 4 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα δύο container. Από τον πίνακα 4.9 διαπιστώνουμε ότι έχει γίνει βελτίωση του προβλήματος αφού έχει εισαχθεί καινούριο πακέτο με μεταβολές στον υπολειπόμενο χώρο.

4.4.5 ΤΡΙΤΗ ΜΕΘΟΔΟΣ ΒΕΛΤΙΩΣΗΣ

Στην παρούσα ενότητα παρουσιάζεται μια μέθοδος βελτίωσης την οποία δημιούργησα με σκοπό την μεγαλύτερη βελτίωση της αρχικής λύσης συγκριτικά με την πρώτη μέθοδο βελτίωσης και την δεύτερη μέθοδο βελτίωσης. Η κεντρική ιδέα του αλγορίθμου είναι η μεταφορά ενός πακέτου από τα πακέτα με την μεγαλύτερη ζήτηση από το container με τον περισσότερο ελεύθερο χώρο στο αμέσως επόμενο container από πλευράς ελεύθερου χώρου. Έτσι απελευθερώνεται χώρος για την προσπάθεια εισαγωγής καινούριου πακέτου.

Algorithm 6 Improvement heuristic 3

Require: $G = \text{containers, remainingblocks, nofdrones, chargingtimetofull}$

procedure IMPROVEMENT 3(G, T)

2: *sort the containers by the remainingspace in descending order*

4: **while** number of remaining blocks > 0 and the total remaining space of containers \geq the size of the new blocks to stack **do**

6: **for** $i \leftarrow 0$ to blocks of the closedrone 0 **do**

8: **if** the i th name == the max name **then**

get the i th block

Break

end if

10: **end for**

12: **if** the remaining space of the 2nd container $>$ the total length of the i th block **then**

Get the i th block from the container 1

Add the i th block to the 2nd container

Create a power set made out of delivery blocks in the container, and select a subset whose summed battery consumption is most similar to the battery amount

Move blocks in the select a subset to the end of the container. Remove the charging

Calculate the total length of the container

if total length \leq operation time **then**

18: **if** the total length of the first remaining block is lower than the closedrone 1 remaining space **then**

Add a remaining block to the closedrone 1

20: *Create a power set made out of delivery blocks in the container, and select a subset whose summed battery consumption is most similar to the battery amount*

Move blocks in the select a subset to the end of the container. Remove the charging

22: *Calculate the total length of the container*

if total length $>$ operation time **then**

24: *Withdraw lines [21,20,19,18]*

Break

26: **else**

Pass

28: **end if**

30: **else**

Break

end if

32: **else**

Get the i th block from the container 2

34: *Add the i th block to the 1st container*

Break and terminate

36: **end if**

38: **else**

Break

end if

40: **end while**

end procedure

4.4.6 ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για να επιτύχουμε την καλύτερη δυνατή επίδειξη του improvement 3 αλγόριθμου, θα χρησιμοποιήσουμε πρόβλημα της κατηγορίας medium. Τα προβλήματα της κατηγορίας medium είναι προβλήματα που αποτελούνται από 4 drones και 20 προορισμούς (4 drones and 20 destinations). Σκοπός του αλγόριθμου είναι από το container με τον περισσότερο ελεύθερο χώρο να παίρνουμε το block με την μεγαλύτερη ζήτηση. Έπειτα ελέγχουμε αν μπορούμε να το μεταφέρουμε στο επόμενο container. Τα container ταξινομούνται από το μεγαλύτερο στο μικρότερο όσο αφορά τον ελεύθερο χώρο τους. Αν όλα αυτά είναι ικανά να γίνουν γίνεται προσπάθεια εισαγωγής block από αυτά που περισσεύουν, αλλιώς ο αλγόριθμος επιστρέφει το block στο αρχικό container και τερματίζει. Αν το block που περισσεύει εισαχθεί κανονικά ο αλγόριθμος υπολογίζει τον συνολικό υπολειπόμενο χώρο και ξαναταξινομεί τα container με τον ίδιο τρόπο και ξαναξεκινάει, αλλιώς επιστρέφει τα blocks στην θέση τους και τερματίζει. Για περισσότερες πληροφορίες ανατρέξτε στον Αλγόριθμο Improvement 3. Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται για την συνάρτηση shaking της BVNS.

NAME OF DRONE	REMAINING SPACE	CUSTOMER										
0	0.261	3	7	0	5	5	1	10	10	12	19	11
1	0.06	12	7	0	5	2	10	1	8	19	3	16
2	1.247	3	12	4	2	10	8	19	0	0	5	
3	1.078	3	7	4	2	10	0	5	12	19	11	
	<i>(REMAINING BLOCKS)</i>	6	6	6	6	6	18	18	18	18	18	13
		13	13	13	17	17	15	15	15	9	9	14

ΠΙΝΑΚΑΣ 4.10: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ CONSTRUCTION HEURISTIC 1

NAME OF DRONE	REMAINING SPACE	CUSTOMER										
0	0.261	3	7	0	5	5	1	10	10	12	19	11
1	0.06	12	7	0	5	2	10	1	8	19	3	16
2	0.681	3	4	2	10	8	6	12	0	0	5	
3	0.362	3	7	4	2	10	0	5	12	11	19	19
	<i>(REMAINING BLOCKS)</i>	6	6	6	6	18	18	18	18	18		
		13	13	13	13	17	17	15	15	15	9	
		9	14									

ΠΙΝΑΚΑΣ 4.11: ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ IMPROVEMENT HEURISTIC 3

Από τον πίνακα 4.10 προκύπτει ότι με την χρήση του αλγόριθμου construction 1 στο container του πρώτου drone πριν αυτό κλείσει εισάγονται 11 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων ο διαθέσιμος χώρος για το container του drone 1 είναι 0.261. Επίσης από τον πίνακα 4.10 μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container του δεύτερου drone. Στο container του δεύτερου drone εισήχθησαν πριν αυτό κλείσει 11 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 2 ο ελεύθερος χώρος που απομένει είναι 0.06. Επίσης

από τον πίνακα **4.10** μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container του τρίτου drone. Στο container του τρίτου drone εισήχθησαν πριν αυτό κλείσει 10 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 3 ο ελεύθερος χώρος που απομένει είναι 1.247. Επίσης από τον πίνακα **4.10** μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου construction 1 στο container του τέταρτου drone. Στο container του τέταρτου drone εισήχθησαν πριν αυτό κλείσει 10 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 4 ο ελεύθερος χώρος που απομένει είναι 1.078. Τέλος από τον πίνακα **4.10** βλέπουμε ότι με την χρήση του αλγόριθμου construction 1 υπάρχουν 22 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα τέσσερα container.

Από τον πίνακα **4.11** προκύπτει ότι με την χρήση του αλγόριθμου improvement 3 στο container του πρώτου drone έχουν εισαχθεί 11 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων ο ελεύθερος χώρος που απομένει είναι στο container του drone 1 είναι 0.261. Επίσης από τον πίνακα **4.11** μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου improvement 3 στο container του δεύτερου drone. Στο container του δεύτερου drone έχουν εισαχθεί 11 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 2 ο ελεύθερος χώρος που απομένει είναι 0.06. Επίσης από τον πίνακα **4.11** μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου improvement 3 στο container του τρίτου drone. Στο container του τρίτου drone εισήχθησαν πριν αυτό κλείσει 10 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 3 ο ελεύθερος χώρος που απομένει είναι 0.681. Επίσης από τον πίνακα **4.11** μπορούμε να εξάγουμε πληροφορίες και τον τρόπο εισαγωγής των πακέτων με την χρήση του αλγόριθμου improvement 3 στο container του τέταρτου drone. Στο container του τέταρτου drone εισήχθησαν πριν αυτό κλείσει 11 από τα 64 συνολικά πακέτα. Με την εισαγωγή αυτών των πακέτων στο container του drone 4 ο ελεύθερος χώρος που απομένει είναι 1.078. Τέλος από τον πίνακα **4.11** βλέπουμε ότι με την χρήση του αλγόριθμου improvement 3 υπάρχουν 21 πακέτα τα οποία δεν έχουν εισαχθεί σε κανένα από τα τέσσερα container.

ΚΕΦΑΛΑΙΟ 5

ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕΛΕΤΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

5.1 ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕΛΕΤΗ

Για την διενέργεια των πειραμάτων δημιουργήθηκαν στιγμιότυπα προβλημάτων τριών διαφορετικών μεγεθών: τα πρόβλημα μικρού μεγέθους (small) με την χρήση 2 drones και 10 προορισμούς (2 drones and 10 destinations), τα πρόβλημα μεσαίου μεγέθους (medium) με την χρήση 4 drones και 20 προορισμούς (4 drones and 20 destinations) και τα πρόβλημα μεγάλου μεγέθους (large) με 10 drones και 50 προορισμούς (10 drones and 50 destinations). Για το κάθε μέγεθος δημιουργήθηκαν άλλα 20 τυχαία διαφορετικά προβλήματα. Για το κάθε πρόβλημα η απόσταση και η καθημερινή ζήτηση πακέτων δημιουργήθηκε με την χρήση τυχαίων πραγματικών αριθμών από το 1 μέχρι το 10 για την απόσταση και με την χρήση ακεραίων αριθμών από το 1 μέχρι το 5 για την καθημερινή ζήτηση πακέτων αντίστοιχα.

Για την διεκπεραίωση των πειραμάτων χρησιμοποιήθηκε ένας προσωπικός υπολογιστής με έναν Intel®Core™i5 2.80 GHz 8ης γενιάς και 8 GB RAM και λειτουργικό σύστημα Windows 10 64-bit. Αν και τα πειράματα δεν λύθηκαν με δυνατότητα χρήσης πολλαπλών πυρήνων, αλλά για την διασφάλιση της μέγιστης απόδοσης να είναι η μόνη διεργασία πέρα από τις αναγκαίες του λειτουργικού συστήματος που τρέχει. Επιπροσθέτως χρησιμοποιήθηκε η Python 3.8.

5.2 ΑΠΟΤΕΛΕΣΜΑΤΑ

Size	# Drones	# Destinations	Obj Function	construction 1	construction 2	construction 3
Small0	2	10	33	8	11	11
Small1	2	10	24	4	6	5
Small2	2	10	29	7	12	10
Small3	2	10	34	14	20	19
Small4	2	10	22	2	4	2
Small5	2	10	33	12	17	12
Small6	2	10	25	4	7	4
Small7	2	10	29	10	15	13
Small8	2	10	29	9	13	12
Small9	2	10	25	6	8	7
Small10	2	10	34	12	18	14
Small11	2	10	26	6	8	8
Small12	2	10	27	7	11	10
Small13	2	10	30	10	16	12
Small14	2	10	34	12	18	16
Small15	2	10	25	7	11	7
Small16	2	10	26	4	8	6
Small17	2	10	29	7	13	10
Small18	2	10	36	12	20	17
Small19	2	10	33	11	19	13
Avg			32,42	8,2	12,75	10,4

ΠΙΝΑΚΑΣ 5.1: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ

Στον 5.1 πίνακα βλέπουμε μία σύνοψη των αποτελεσμάτων από τα προβλήματα που ανήκουν στην κατηγορία μικρού μεγέθους small τα οποία αποτελούνται από 2 drones και έχουν 10 destinations μετά την χρήση των construction heuristics. Σε αυτόν τον πίνακα αντλούμε πληροφορίες όπως το όνομα του κάθε προβλήματος τον αριθμό των drones την αντικειμενική συνάρτηση και τον αριθμό από πακέτα που περισσεύουν χρησιμοποιώντας την κάθε μία από τις construction heuristic συναρτήσεις. Τα ονόματα των προβλημάτων είναι small0, small1 και φτάνει μέχρι small19. Ο μέσος όρος αντικειμενικής συνάρτησης σε αυτά τα προβλήματα είναι 32.42. Με μικρότερο πρόβλημα της κατηγορίας σύμφωνα με την αντικειμενική συνάρτηση είναι το small4 με 22 πακέτα και μεγαλύτερο το small18 με 36 πακέτα. Παρατηρείται επίσης ότι στα προβλήματα μικρού μεγέθους (small) από όλες τις construction heuristics συναρτήσεις η construction heuristic 1 έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η construction heuristic 1 έχει μέσο όρο 8.2 πακέτα που περισσεύουν, η construction heuristic 2 έχει μέσο όρο 12,75 πακέτα που περισσεύουν και η construction 3 έχει μέσο όρο 10.4. Στην construction heuristic 1 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο small4 με 2, ενώ στο small3 περισσεύουν 14 πακέτα που είναι τα περισσότερα. Στην construction heuristic 2 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο small4 με 4, ενώ στο small3 και στο small18 περισσεύουν 20 πακέτα

που είναι τα περισσότερα. Στην construction heuristic 3 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο small4 με 2, ενώ στο small3 περισσεύουν 19 πακέτα.

Size	# Drones	# Destinations	Obj Function	vnd	bvns
Small0	2	10	33	8	8
Small1	2	10	24	3	4
Small2	2	10	29	7	7
Small3	2	10	34	13	14
Small4	2	10	22	2	2
Small5	2	10	33	12	12
Small6	2	10	25	4	4
Small7	2	10	29	9	9
Small8	2	10	29	9	9
Small9	2	10	25	6	6
Small10	2	10	34	12	12
Small11	2	10	26	6	6
Small12	2	10	27	7	7
Small13	2	10	30	10	10
Small14	2	10	34	11	12
Small15	2	10	25	6	6
Small16	2	10	26	4	4
Small17	2	10	29	7	7
Small18	2	10	36	12	12
Small19	2	10	33	11	11
Avg			32,42857143	7,95	8,1

ΠΙΝΑΚΑΣ 5.2: ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ

Στον **5.2** πίνακα έχουμε τα αποτελέσματα της χρήσης BVNS και VND στα προβλήματα κατηγορίας μικρού μεγέθους (small) μετά από την κατασκευή αρχικής λύσης με την construction heuristic 1 όπου αποδείχτηκε η καλύτερη construction heuristic για τα προβλήματα κατηγορίας μικρού μεγέθους (small). Παρατηρείται ότι στα προβλήματα μικρού μεγέθους (small) η VND έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η VND έχει μέσο όρο 7.95 πακέτα που περισσεύουν, η BVNS έχει μέσο όρο 8.1 πακέτα που περισσεύουν. Με την χρήση της VND το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο small4 με 2, ενώ στο small3 περισσεύουν 13 πακέτα που είναι τα περισσότερα. Με την χρήση της BVNS το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο small4 με 2, ενώ στο small3 περισσεύουν 14 πακέτα που είναι τα περισσότερα.

Size	# Drones	# Destinations	Obj Function	construction 1	construction 2	construction 3
Medium0	4	20	59	19	28	26
Medium1	4	20	73	29	45	39
Medium2	4	20	66	23	38	35
Medium3	4	20	65	21	35	30
Medium4	4	20	64	22	35	29
Medium5	4	20	56	16	28	24
Medium6	4	20	56	16	24	23
Medium7	4	20	57	17	27	23
Medium8	4	20	67	27	43	36
Medium9	4	20	67	28	39	34
Medium10	4	20	54	17	26	24
Medium11	4	20	63	20	35	31
Medium12	4	20	51	9	15	13
Medium13	4	20	60	23	32	28
Medium14	4	20	57	17	25	21
Medium15	4	20	60	20	32	28
Medium16	4	20	52	12	20	15
Medium17	4	20	58	14	22	21
Medium18	4	20	54	14	22	20
Medium19	4	20	64	27	36	32
Avg			66,125	19,55	30,35	26,6

ΠΙΝΑΚΑΣ 5.3: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ

Στον **5.3** πίνακα βλέπουμε μία σύνοψη των αποτελεσμάτων από τα προβλήματα που ανήκουν στην κατηγορία μεσαίου μεγέθους medium τα οποία αποτελούνται από 4 drones και έχουν 20 destinations. Σε αυτόν τον πίνακα αντλούμε πληροφορίες όπως το όνομα του κάθε προβλήματος, τον αριθμό των drones, την αντικειμενική συνάρτηση και τον αριθμό από πακέτα που περισσεύουν, χρησιμοποιώντας την κάθε μία από τις construction heuristic συναρτήσεις. Τα ονόματα των προβλημάτων είναι medium0, medium1 και φτάνει μέχρι medium19. Ο μέσος όρος αντικειμενικής συνάρτησης της κατηγορίας σε αυτά τα προβλήματα είναι 66.125. Με μικρότερο πρόβλημα σύμφωνα με την αντικειμενική συνάρτηση είναι το medium12 με 51 πακέτα και μεγαλύτερο το medium1 με 73 πακέτα. Παρατηρείται επίσης ότι στα medium προβλήματα από όλες τις construction heuristic συναρτήσεις η construction heuristic 1 έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η construction heuristic 1 έχει μέσο όρο 19.55 πακέτα που περισσεύουν, η construction heuristic 2 έχει μέσο όρο 30.35 πακέτα που περισσεύουν και η construction heuristic 3 έχει μέσο όρο 26.6. Στην construction heuristic 1 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο medium12 με 9, ενώ στο medium1 περισσεύουν 29 πακέτα που είναι τα περισσότερα. Στην construction heuristic 2 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο medium12 με 15, ενώ στο medium1 περισσεύουν 45 πακέτα που είναι τα περισσότερα. Στην construction heuristic 3 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο medium12 με 13, ενώ στο medium1 περισσεύουν 39 πακέτα που είναι τα περισσότερα.

Size	# Drones	# Destinations	Obj Function	vnd	bvns
Medium0	4	20	59	19	19
Medium1	4	20	73	29	29
Medium2	4	20	66	23	22
Medium3	4	20	65	21	21
Medium4	4	20	64	21	21
Medium5	4	20	56	15	14
Medium6	4	20	56	16	16
Medium7	4	20	57	17	16
Medium8	4	20	67	26	25
Medium9	4	20	67	28	27
Medium10	4	20	54	16	15
Medium11	4	20	63	20	19
Medium12	4	20	51	8	8
Medium13	4	20	60	23	22
Medium14	4	20	57	17	16
Medium15	4	20	60	19	18
Medium16	4	20	52	12	10
Medium17	4	20	58	13	14
Medium18	4	20	54	14	14
Medium19	4	20	64	27	26
Avg			66,125	19,2	18,6

ΠΙΝΑΚΑΣ 5.4: ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ

Στον 5.4 πίνακα έχουμε τα αποτελέσματα της χρήσης BVNS και VND στα προβλήματα της κατηγορίας μεσαίου μεγέθους (medium) μετά από την κατασκευή αρχικής λύσης με την construction 1 όπου αποδείχτηκε η καλύτερη construction heuristic για τα προβλήματα της κατηγορίας μεσαίου μεγέθους (medium). Παρατηρείται ότι στα προβλήματα μεσαίου μεγέθους (medium) η BVNS έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η VND έχει μέσο όρο 19.2 πακέτα που περισσεύουν, η BVNS έχει μέσο όρο 18.6 πακέτα που περισσεύουν. Με την χρήση της VND το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο medium12 με 8, ενώ στο medium1 περισσεύουν 29 πακέτα που είναι τα περισσότερα. Με την χρήση της BVNS το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο medium12 με 8, ενώ στο medium1 περισσεύουν 29 πακέτα που είναι τα περισσότερα.

Size	# Drones	# Destinations	Obj Function	construction 1	construction 2	construction 3
Large0	10	50	165	65	95	88
Large1	10	50	153	48	73	67
Large2	10	50	154	49	74	68
Large3	10	50	163	63	93	87
Large4	10	50	150	40	70	71
Large5	10	50	142	50	72	75
Large6	10	50	144	44	64	59
Large7	10	50	147	42	67	61
Large8	10	50	135	36	55	55
Large9	10	50	140	36	60	58
Large10	10	50	146	47	76	70
Large11	10	50	140	40	60	54
Large12	10	50	144	34	63	58
Large13	10	50	148	52	78	72
Large14	10	50	157	62	87	81
Large15	10	50	143	43	63	57
Large16	10	50	140	40	60	55
Large17	10	50	178	68	108	101
Large18	10	50	138	38	58	53
Large19	10	50	141	39	61	56
Avg			148,4	46,8	71,85	67,3

ΠΙΝΑΚΑΣ 5.5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΩΝ CONSTRUCTION HEURISTICS ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ

Στον 5.5 πίνακα βλέπουμε μία σύνοψη των αποτελεσμάτων από τα προβλήματα που ανήκουν στην κατηγορία μεγάλου μεγέθους large τα οποία αποτελούνται από 10 drones και έχουν 50 destinations. Σε αυτόν τον πίνακα αναλύουμε πληροφορίες, όπως το όνομα του κάθε προβλήματος, τον αριθμό των drones, την αντικειμενική συνάρτηση και τον αριθμό από πακέτα που περισσεύουν χρησιμοποιώντας την κάθε μία από τις construction heuristic συναρτήσεις. Τα ονόματα των προβλημάτων είναι large0, large1 και φτάνει μέχρι large19. Ο μέσος όρος αντικειμενικής συνάρτησης της κατηγορίας σε αυτά τα προβλήματα είναι 148.4. Με μικρότερο πρόβλημα σύμφωνα με την αντικειμενική συνάρτηση είναι το large18 με 138 πακέτα και μεγαλύτερο το large17 με 178 πακέτα. Παρατηρείται επίσης ότι στα προβλήματα μεγάλου μεγέθους (large) από όλες τις construction heuristic συναρτήσεις η construction heuristic 1 έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η construction heuristic 1 έχει μέσο όρο 46.8 πακέτα που περισσεύουν, η construction heuristic 2 έχει μέσο όρο 71.85 πακέτα που περισσεύουν και η construction heuristic 3 έχει μέσο όρο 67.3. Στην construction heuristic 1 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο large12 με 34, ενώ στο large17 περισσεύουν 68 πακέτα που είναι τα περισσότερα. Στην construction heuristic 2 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο large8 με 55, ενώ στο large17 περισσεύουν 108 πακέτα που είναι τα περισσότερα. Στην construction heuristic 3 το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο large18 με 53, ενώ στο large17 περισσεύουν 101 πακέτα.

Size	# Drones	# Destinations	Obj Function	vnd	bvns
Large0	10	50	165	64	62
Large1	10	50	153	47	46
Large2	10	50	154	48	46
Large3	10	50	163	63	63
Large4	10	50	150	40	40
Large5	10	50	142	49	48
Large6	10	50	144	43	42
Large7	10	50	147	41	40
Large8	10	50	135	36	36
Large9	10	50	140	35	33
Large10	10	50	146	47	47
Large11	10	50	140	39	38
Large12	10	50	144	34	34
Large13	10	50	148	51	50
Large14	10	50	157	61	61
Large15	10	50	143	43	43
Large16	10	50	140	39	38
Large17	10	50	178	68	68
Large18	10	50	138	38	38
Large19	10	50	141	38	38
Avg			148,4	46,2	45,55

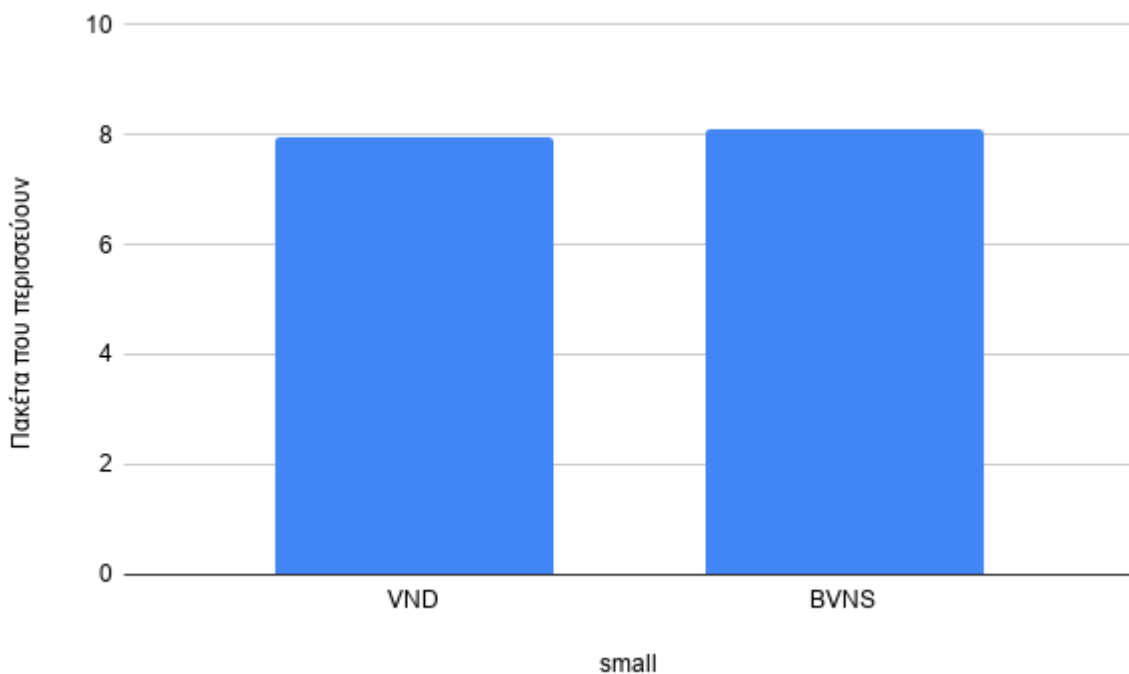
ΠΙΝΑΚΑΣ 5.6: ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ

Στον 5.6 πίνακα έχουμε τα αποτελέσματα της χρήσης BVNS και VND στα προβλήματα κατηγορίας μεγάλου μεγέθους (large) μετά από την κατασκευή αρχικής λύσης με την construction heuristic 1 όπου αποδείχτηκε η καλύτερη construction heuristic για τα προβλήματα κατηγορίας. Παρατηρείται ότι στα προβλήματα μεγάλου μεγέθους (large) η BVNS έχει τα καλύτερα αποτελέσματα αφού έχει κατά μέσο όρο τα λιγότερα πακέτα που περισσεύουν. Η VND έχει μέσο όρο 46.2 πακέτα που περισσεύουν, η BVNS έχει μέσο όρο 45.55 πακέτα που περισσεύουν. Με την χρήση της VND το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο large12 με 34, ενώ στο large17 περισσεύουν 68 πακέτα που είναι τα περισσότερα. Με την χρήση της BVNS το πρόβλημα με τα λιγότερα πακέτα που περισσεύουν είναι στο large9 με 33, ενώ στο large17 περισσεύουν 68 πακέτα που είναι τα περισσότερα.

5.3 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND

Με δεδομένο ότι η εφαρμογή των μεθόδων BVNS και VND πραγματοποιήθηκαν πάνω στα ίδια ακριβώς προβλήματα, δηλαδή στα 20 πρόβλημα μικρού μεγέθους (small), στα 20 πρόβλημα μεσαίου μεγέθους (medium) και στα 20 πρόβλημα μεγάλου μεγέθους (large). Αρχικά, πρέπει να επισημάνουμε ότι η επιλογή ίδιου τρόπου αναπαράστασης λύσης μας επιτρέπει να συγκρίνουμε τις δύο μεθόδους καθώς εξαιτίας αυτού του γεγονότος, οι δύο αλγόριθμοι εξερεύνησαν τον ίδιο χώρο αναζήτησης. Έπειτα αξίζει να σημειωθεί ότι η BVNS χρησιμοποιεί μία shaking μέθοδο η οποία είναι μια ακόμα διαφορά της με την VND.

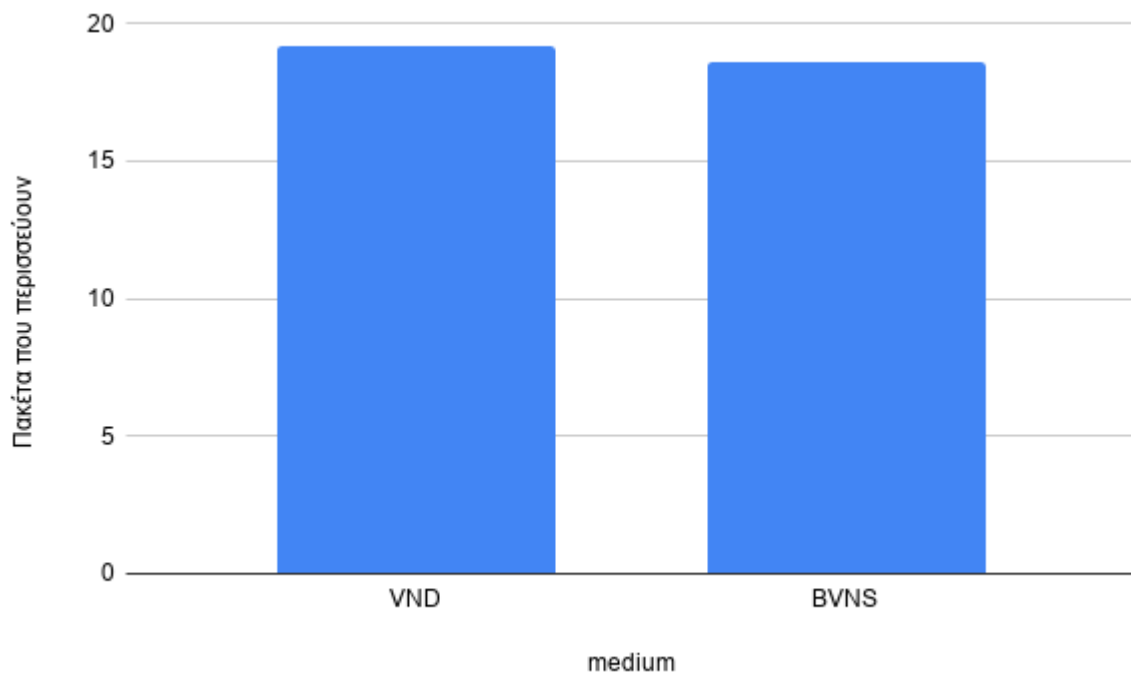
5.3.1 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ



ΣΧΗΜΑ 5.3.1: ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ

Στα προβλήματα μικρού μεγέθους (small) συγκρίνοντας τους μέσους όρους πακέτων που περισσεύουν, ως καλύτερη μέθοδος από την εικόνα 5.3.1 προκύπτει ότι είναι η VND μεθοδολογία, αφού περισσεύουν λιγότερα πακέτα.

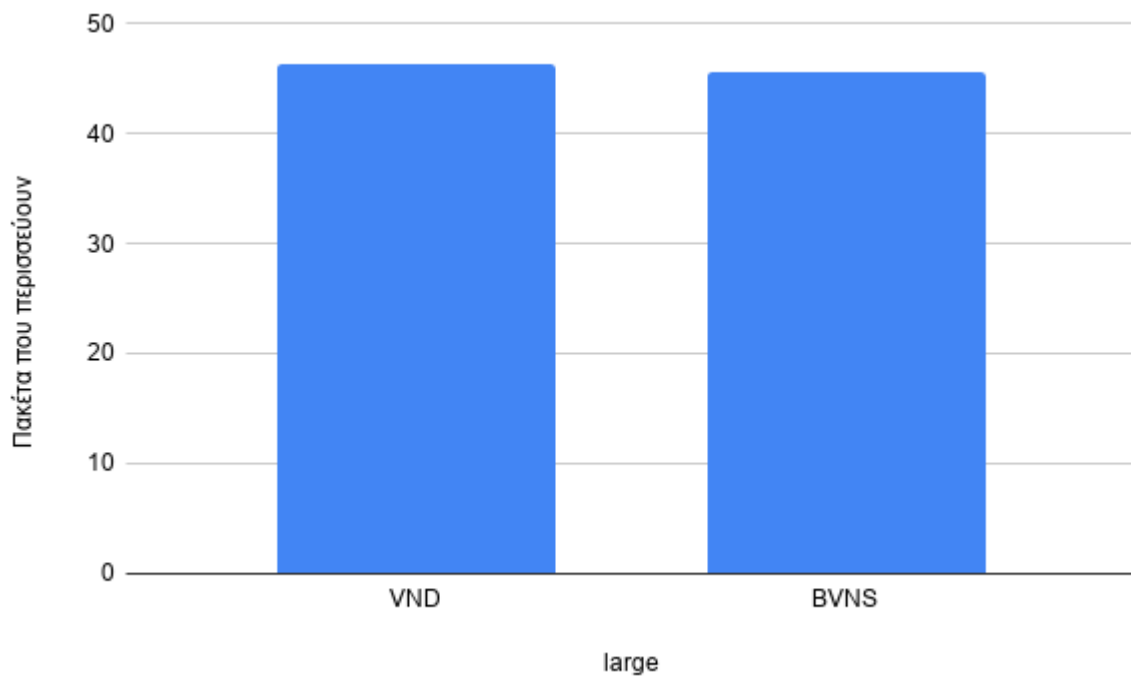
5.3.2 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ



ΣΧΗΜΑ 5.3.2: ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΣΑΙΟΥ ΜΕΓΕΘΟΥΣ

Στα προβλήματα μεσαίου μεγέθους (medium) συγκρίνοντας τους μέσους όρους πακέτων που περισσεύουν, ως καλύτερη μέθοδος από την εικόνα 5.3.2 προκύπτει ότι είναι η BVNS μεθοδολογία, αφού περισσεύουν λιγότερα πακέτα.

5.3.3 ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ



ΣΧΗΜΑ 5.3.3: ΣΥΓΚΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ BVNS ΚΑΙ VND ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ

Στα προβλήματα μεγάλου μεγέθους (large) συγκρίνοντας τους μέσους όρους πακέτων που περισσεύουν, ως καλύτερη μέθοδος από την εικόνα 5.3.3 προκύπτει ότι είναι η BVNS μεθοδολογία, αφού περισσεύουν λιγότερα πακέτα.

ΚΕΦΑΛΑΙΟ 6

ΕΠΙΛΟΓΟΣ

Η διπλωματική αυτή εργασία που πραγματοποιήθηκε στο πλαίσιο των μεταπτυχιακών μου σπουδών στο Τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας και γίνεται η ανάλυση και προσπάθεια εύρεσης λύσης με την χρήση της αναζήτησης μεταβαλλόμενης γειτονιάς (VNS) σε πρόβλημα της κατηγορίας VRP.

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στόχος της εργασίας ήταν η επίλυση ενός προβλήματος της κατηγορίας VRP με την χρήση της VNS μεθωρετικής και μίας παραλλαγής της, την VND. Για την καλύτερη κατανόηση του προβλήματος πραγματοποιήθηκε θεωρητική μελέτη. Για την επίλυση του προβλήματος δημιουργήθηκαν τρεις αλγόριθμοι κατασκευής και τρεις αλγόριθμοι βελτίωσης. Μετά την εκτέλεση των αλγορίθμων στα στιγμιότυπα που δημιουργήθηκαν διαπιστώνουμε ότι ο πρώτος αλγόριθμος κατασκευής αρχικής λύσης βρίσκει κατά μέσο όρο καλύτερη αρχική λύση συγκριτικά με τους υπόλοιπους δύο αλγόριθμους, έχοντας ως αντικειμενική συνάρτηση την μεγιστοποίηση των πακέτων που μεταφέρονται. Τέλος βλέπουμε ότι η VND παρουσιάζει καλύτερα αποτελέσματα για τα small προβλήματα ενώ η BVNS παρουσιάζει καλύτερα αποτελέσματα για τα medium και large προβλήματα.

6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στο πλαίσιο αυτής της διπλωματικής εργασίας αναπτύχθηκε σε γλώσσα προγραμματισμού Python μία υλοποίηση της VNS μεθωρετικής μεθόδου για την επίλυση VRP προβλήματος. Στο συγκεκριμένο πρόβλημα ως οχήματα χρησιμοποιήθηκαν drones. Η έρευνα πάνω σε VRP προβλήματα με χρήση drones αυξάνεται ραγδαία. Η επίλυση του συγκεκριμένου προβλήματος μπορεί να φανεί χρήσιμη σε εταιρίες όπως η Amazon <https://www.amazon.com/>, η οποία δραστηριοποιείται σε μεταφορές διαφόρων αντικειμένων όλων των μεγεθών ανά τον κόσμο. Για τις μεταφορές αυτές ή για ένα κομμάτι αυτών θα μπορούσε να χρησιμοποιήσει drones. Στην προκειμένη περίπτωση το συγκεκριμένο πρόβλημα μπορεί να επεκταθεί στην παράδοση πακέτων σε συγκεκριμένα χρονικά

πλαίσια. Ενδιαφέρον παρουσιάζει η ενδεχόμενη υλοποίηση του προβλήματος ώστε να λαμβάνονται υπόψη οι καιρικές συνθήκες αφού μιλάμε για real - world πρόβλημα, επειδή τα περισσότερα drones δεν είναι ικανά να ανταπεξέλθουν σε αντίξοες καιρικές συνθήκες και γι αυτό δεν λαμβάνονται υπόψη στην παρούσα εργασία. Ένας ακόμα παράγοντας που πρέπει να ληφθεί υπόψη είναι η ανθρώπινη παρέμβαση. Σε περίπτωση που δεν είναι δυνατό να παραλάβει κάποιος άνθρωπος το οποιοδήποτε πακέτο σημαίνει ότι θα υπάρξουν καθυστερήσεις στην παράδοση. Ένας ακόμα λόγος για να γίνουν μεταφορές χωρίς ανθρώπινη παρέμβαση εκτός από τις καθυστερήσεις είναι και η πανδημία του COVID19 ταλανίζει ακόμα ολόκληρο τον πλανήτη. <https://www.who.int/emergencies/diseases/novel-coronavirus-2019> Αυτό θα έχει ως αποτέλεσμα την ασφαλέστερη μεταφορά και την βοήθεια στον περιορισμό του ιού. Έτσι με την χρήση των drones θα βελτιωθεί η καθημερινότητα των ανθρώπων. <https://blog.werobotics.org/2020/04/25/cargo-drones-covid-19/> Επιπροσθέτως θα μπορούσε να γίνει υλοποίηση με περισσότερα από ένα depot. Επίσης ενδιαφέρον θα παρουσίαζε και λύση του παρόντος προβλήματος με την χρήση λιγότερων drones (Poikonen and Campbell 2021). Τέλος για την υλοποίηση των προτάσεων αυτών πρέπει να ληφθούν υπόψη θέματα όπως αποτελεσματικότητα, ασφάλεια και προσωπικά δεδομένα. Σημαντικό είναι να βελτιώσουμε τις παρούσες λύσεις. Επίσης τα ευαίσθητα προσωπικά δεδομένα πρέπει να προστατευτούν από διάφορους επιτήδειους που θα προσπαθήσουν να επωφεληθούν.

ΑΝΑΦΟΡΕΣ

- Aras, N., D. Aksen, and M. Tuğrul Tekin (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C: Emerging Technologies* 19(5), 866–884.
- Avdoshin, S. and E. Beresneva (2019). Constructive heuristics for capacitated vehicle routing problem: a comparative study. *Proceedings of the Institute for System Programming of the RAS* 31, 145–156.
- Baldacci, R., N. Christofides, and A. Mingozzi (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* 115, 351–385.
- Bouhmala, N., K. Hjelmervik, and K. I. Øvergaard (2015). A generalized variable neighborhood search for combinatorial optimization problems. *Electronic Notes in Discrete Mathematics* 47, 45–52. The 3rd International Conference on Variable Neighborhood Search (VNS'14).
- Brandão, J. (2011). A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers Operations Research* 38(1), 140 – 151. Project Management and Scheduling.
- Burke, E. K. and G. Kendall (2013). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (2nd ed.). Springer Publishing Company, Incorporated.
- Cacchiani, V., V. Hemmelmayr, and F. Tricoire (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics* 163, 53–64.
- Chaudhuri, A. and K. De (2011). Fuzzy multi-objective linear programming for traveling salesman problem. *African Journal of Mathematics and Computer Science Research* 4, 64–70.
- Chen, P., H. kuan Huang, and X.-Y. Dong (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications* 37(2), 1620–1627.
- Cordeau, J.-F. and G. Laporte (2007). The dial-a-ride problem (darp): Models and algorithms. *Annals OR* 153, 29–46.

- Cordeau, J.-F., G. Laporte, and A. Mercier (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52(8), 928–936.
- Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations Research* 6(6), 791–812.
- D’Andrea, R. (2014). Guest editorial can drones deliver? *IEEE Transactions on Automation Science and Engineering* 11(3), 647–648.
- Dantzig, G. B. and J. H. Ramser (1959). The truck dispatching problem. *Management Science* 6(1), 80–91.
- Deroussi, L. (2016a). *An Introduction to Metaheuristics*, Chapter 3, pp. 37–55. John Wiley Sons, Ltd.
- Deroussi, L. (2016b). *A Review of Logistic Problems*, Chapter 2, pp. 13–36. John Wiley Sons, Ltd.
- Dorling, K., J. Heinrichs, G. G. Messier, and S. Magierowski (2017). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(1), 70–85.
- D’Souza, C., O. S N, and J. Senthilnath (2012). Pickup and delivery problem using metaheuristics techniques. *Expert Systems with Applications* 39, 328–334.
- Duarte, A., N. Mladenović, J. Sánchez-Oro Calvo, and R. Todosijević (2018). *Variable neighborhood descent*, pp. 341–367.
- Duhamel, C., P. Lacomme, and C. Prodhon (2012). A hybrid evolutionary local search with depth first search split procedure for the heterogeneous vehicle routing problems. *Engineering Applications of Artificial Intelligence* 25, 345–358.
- Eksioglu, B., A. V. Vural, and A. Reisman (2009). The vehicle routing problem: A taxonomic review. *Computers Industrial Engineering* 57(4), 1472 – 1483.
- Ferrandez, S. M., T. Harbison, T. Weber, R. Sturges, and R. Rich (2016). Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management* 9(2).
- Fleszar, K., I. H. Osman, and K. S. Hindi (2009). A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research* 195(3), 803–809.
- Gajpal, Y., V. Roy, and B. Sahay (2019). Vehicle routing for a mid-day meal delivery distribution system. *Heliyon* 5, e01158.

- Gendreau, M. and J.-Y. Potvin (Eds.) (2019). *Handbook of Metaheuristics*. Number 978-3-319-91086-4 in International Series in Operations Research and Management Science. Springer.
- Ghannadpour, S. F. and M. Hooshfar (2016). Multi-objective vehicle routing problem with time windows and fuel consumption minimizing. In *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems - Volume 1: ICORES*, pp. 92–99. INSTICC: SciTePress.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers Operations Research* 13(5), 533–549. Applications of Integer Programming.
- Goksal, F., I. Karaoglan, and F. Altiparmak (2013). A hybrid particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers Industrial Engineering* 65, 3953.
- Ha, Q. M., Y. Deville, Q. D. Pham, and M. H. H (2018). On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies* 86, 597 – 621.
- Hansen, P. and N. Mladenović (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130(3), 449 – 467.
- Hansen, P., N. Mladenović, and J. M. Pérez (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research* 175(1), 367–407.
- Hansen, P., N. Mladenović, and D. Perez-Britos (2001). Variable neighborhood decomposition search. *Journal of Heuristics* 7, 335–350.
- Hertz, A. and M. Mittaz (2001). A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. *Transportation Science* 35, 425–434.
- Ho, S. and D. Haugland (2004). A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers Operations Research* 31, 1947–1964.
- Holland, J. H. (1984). *Genetic Algorithms and Adaptation*, pp. 317–333. Boston, MA: Springer US.
- Hong, L. (2012). An improved lns algorithm for real-time vehicle routing problem with time windows. *Computers OR* 39, 151–163.
- Jünger, M., G. Reinelt, and G. Rinaldi (1995). Chapter 4 the traveling salesman problem. In *Network Models*, Volume 7 of *Handbooks in Operations Research and Management Science*, pp. 225–330. Elsevier.
- Kim, J., H. Moon, and H. Jung (2020). Drone-based parcel delivery using the rooftops of city buildings: Model and solution. *Applied Sciences* 10(12).
- Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science* 220(4598), 671–680.

- Kuo, Y. and C.-C. Wang (2012). A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications* 39(8), 6949-6954.
- Labadie, N., R. Mansini, J. Melechovský, and R. Calvo (2012). The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research* 220, 1527.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science* 43, 408–416.
- Laporte, G. and F. Semet (2002). Classical heuristics for the capacitated vrp. In *The vehicle routing problem*, pp. 109–128. SIAM.
- Lei, H., G. Laporte, and B. Guo (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers OR* 38, 1775–1783.
- Li, X., S. C. Leung, and P. Tian (2012). A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. *Expert Systems with Applications* 39(1), 365 – 374.
- Liao, T.-Y. (2004). Tabu search algorithm for dynamic vehicle routing problems under real-time information. *Transportation Research Record* 1882(1), 140–149.
- Lin, S.-W. (2013). Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing* 13, 1064–1073.
- Lin, S.-W. and V. F. Yu (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research* 217(1), 94–107.
- Lin, S.-W., V. F. Yu, and C.-C. Lu (2011). A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications* 38(12), 15244-15252.
- Luke, S. (2009). *Essentials of Metaheuristics*. available at <http://cs.gmu.edu/~sean/book/metaheuristics/>.
- Mersheeva, V. and G. Friedrich (2012). Routing for continuous monitoring by multiple micro uavs in disaster scenarios. Volume 242.
- Mladenović, N. (2003). A tutorial on variable neighborhood search. *Les Cahiers du GERAD ISSN 711*, 2440.
- Mladenović, N. and P. Hansen (1997). Variable neighborhood search. *Computers Operations Research* 24(11), 1097 – 1100.
- Naji-Azimi, Z. and M. Salari (2013). A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem. *Applied Mathematical Modelling* 37(6), 4316–4324.

- Nazif, H. and L. S. Lee (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling* 36(5), 2110–2117.
- Osman, I. (1993). Meta-strategy simulated annealing and tabu search algorithms for the vehicle routine problem. *Annals of Operations Research* 41, 421–451.
- Parragh, S. N. and V. Schmid (2013). Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers Operations Research* 40(1), 490 – 497.
- Pessoa, A. A., R. Sadykov, and E. Uchoa (2017). An improved Branch-Cut-and-Price algorithm for heterogeneous vehicle routing problems. In *VeRoLog 2017 - Annual Workshop of the EURO Working Group on Vehicle Routing and Logistics optimization*, Amsterdam, Netherlands.
- Poikonen, S. and J. F. Campbell (2021). Future directions in drone routing research. *Networks* 77(1), 116–126.
- Polacek, M., K. Doerner, R. Hartl, and V. Maniezzo (2008). A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics* 14.
- Pourrahmani, E., M. Delavar, and M. A. Mostafavi (2015). Optimization of an evacuation plan with uncertain demands using fuzzy credibility theory and genetic algorithm. *International Journal of Disaster Risk Reduction* 14.
- Qu, Y. and J. F. Bard (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers Operations Research* 39(10), 2439–2456.
- Rego, C. and C. Roucairol (1996). A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In *Meta-Heuristics*, pp. 661–675. Springer.
- Reinelt, G. (1991). Tsplib traveling salesman problem library. *INFORMS Journal on Computing* 3(4), 376–384.
- Rothlauf, F. (2011). *Design of Modern Heuristics: Principles and Application* (1st ed.). Springer Publishing Company, Incorporated.
- Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). In K. Aardal, G. Nemhauser, and R. Weismantel (Eds.), *Discrete Optimization*, Volume 12 of *Handbooks in Operations Research and Management Science*, pp. 1 – 68. Elsevier.
- Siarry, P. (2016). *Metaheuristics* (1st ed.). Springer Publishing Company, Incorporated.
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*, Volume 74.
- Tavares, L., H. Lopes, and C. Erig Lima (2009). Construction and improvement heuristics applied to the capacitated vehicle routing problem. pp. 690–695.
- Toth, P. and D. Vigo (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing* 15(4), 333–346.

Vidal, T., T. G. Crainic, M. Gendreau, and C. Prins (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers Operations Research* 40, 475–489.

Villegas, J. G., C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco (2011). A grasp with evolutionary path relinking for the truck and trailer routing problem. *Computers Operations Research* 38(9), 1319 – 1334.

Yu, B. and Z. Z. Yang (2011). An ant colony optimization model: The period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review* 47(2), 166–181.