



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Βέλτιστη δρομολόγηση μη
επανδρωμένων οχημάτων σε
τρισδιάστατο χώρο υπό περιορισμούς

Νικόλαος Μπαράς

Επιβλέπων Καθηγητής : Δρ. Μηνάς Δασυγένης
Εργαστήριο Ψηφιακών Συστημάτων και
Αρχιτεκτονικής Υπολογιστών

12 Μαρτίου 2019

Περίληψη

Εξαιτίας της ραγδαίας τεχνολογικής και επιστημονικής ανάπτυξης, τις τελευταίες δεκαετίες παρατηρείται η ολοένα και αυξανόμενη χρήση των αυτόνομων οχημάτων σε πολλούς τομείς, όπως η γεωργία, η εξερεύνηση της θάλασσας και χαρτογράφηση. Παρόλο που τα αυτόνομα οχήματα δίνουν λύση σε μια σειρά από προβλήματα, νέα προβλήματα που χρήζουν λύση έρχονται στην επιφάνεια. Ένα από τα πιο σημαντικά προβλήματα των αυτόνομων οχημάτων είναι ο σχεδιασμός μιας διαδρομής που καλύπτει όλα τα σημεία μιας προκαθορισμένης περιοχής και αποφεύγει σημεία με συγκεκριμένα χαρακτηριστικά, όπως εμπόδια και περιοχές απαγορευμένης πτήσης. Το πρόβλημα αυτό είναι γνωστό και ως Coverage Path Planning (CPP).

Παρόλο που η δρομολόγηση αυτόνομων οχημάτων σε δισδιάστατο χώρο έχει προσελκύσει πολλούς ερευνητές, λίγοι έχουν ασχοληθεί με την δρομολόγηση οχημάτων σε τρισδιάστατο χώρο. Επιπλέον, δεν λαμβάνονται συχνά υπόψη άλλα θέματα όπως η ενέργεια, η ταχύτητα και η χωρητικότητα δεδομένων των οχημάτων.

Λαμβάνοντας υπόψη τη σημασία της βέλτιστης δρομολόγησης αυτόνομων οχημάτων καθώς και την έλλειψη λογισμικού δρομολόγησης, αναπτύξαμε ένα εργαλείο βέλτιστης δρομολόγησης αυτόνομων οχημάτων. Το εργαλείο δέχεται ως είσοδο την περιοχή ενδιαφέροντος σε μορφή τρισδιάστατου πίνακα καθώς και πληροφορίες σχετικά με τα χαρακτηριστικά των οχημάτων (όπως αρχική τοποθεσία, ταχύτητα, συνολική ενέργεια και αποθηκευτικός χώρος). Σαν έξοδο, παράγει την διαδρομή που θα ακολουθήσει το κάθε όχημα προκειμένου να καλυφθεί όλος ο ζητούμενος χώρος. Η παραγόμενη διαδρομή είναι προσαρμοσμένη στο κάθε όχημα βάσει των χαρακτηριστικών του.

Λέξεις κλειδιά: Βέλτιστη δρομολόγηση, αυτόνομο όχημα, δρομολόγηση σε τρισδιάστατο χώρο

Abstract

Due to the rapid technological and scientific development, in recent years, unmanned vehicles are being used in several application domains, such as mapping, agriculture and deep-sea exploration. Although autonomous vehicles provide a solution to a number of problems, new problems that need to be solved are coming to the surface. One of the most important problems of unmanned vehicles is the problem of finding a path that covers the entire area of interest in a predefined environment while avoiding obstacles. This problem is also known as Coverage Path Planning (CPP).

Even though many research works have been focused on solving the CPP problem in 2D environments, the CPP problem in 3D environments has not attracted considerable attention. Furthermore, other issues such as energy, speed and data capacity are often not taken into account.

Taking into account the importance of CPP of autonomous vehicles and the lack of CPP software, we have developed a methodology and a tool for optimal routing of autonomous vehicles. The inputs of the tool include a three dimensional array describing the area of interest as well as information regarding vehicle characteristics (such as starting position, speed, total energy and storage). Our tool produces the route that each vehicle will follow in order to cover the area of interest. The path produced is tailored to each vehicle based on its characteristics.

Keywords: Coverage Path Planning, CPP, autonomous vehicle, 3D Coverage Path Planning

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Βέλτιστη δρομολόγηση μη επανδρωμένων οχημάτων σε τρισδιάστατο χώρο υπο περιορισμούς" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Νικόλαος Μπαράς & Μηνάς Δασυγένης, 2019, Κοζάνη

Υπογραφή Φοιτητή

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον Δρ. Μηνά Δασυγνήνη για την πολύτιμη συμβολή και καθοδήγηση του καθώς και για την υπομονή που έδειξε καθ'όλη την διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αμέριστη ηθική και υλική υποστήριξη που μου παρείχε κατά την διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου που στάθηκαν δίπλα μου και με βοήθησαν με όποιο τρόπο ο καθένας μπορούσε.

Περιεχόμενα

1	Εισαγωγή	12
1.1	Ορισμός του προβλήματος	12
1.2	Κίνητρα και Στόχοι Υλοποίησης	15
1.3	Περιπτώσεις παρόμοιων ερευνητικών έργων	16
1.4	Διάρθρωση κειμένου	17
2	Θεωρητικό υπόβαθρο	18
2.1	C-Sharp	18
2.2	Αλγόριθμος αναζήτησης	19
2.3	Αλγόριθμος Coverage Path Planning (CPP)	20
2.4	Γράφημα	20
2.5	Δενδρική δομή	21
2.6	Συνδεδεμένο στοιχείο (Connected Component)	22
2.7	Ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree)	22
2.8	Νήματα (threads)	24
2.9	Εργαλεία που χρησιμοποιήθηκαν	24
2.9.1	Visual Studio	24
2.9.2	Visual Studio Code	25
2.9.3	Notepad++	26
2.9.4	Git	26
2.10	Σύνοψη Κεφαλαίου	27
3	Υλοποίηση του λογισμικού μέρους	28
3.1	Γενική επισκόπηση του λογισμικού	28
3.2	Περιγραφή του περιβάλλοντος	29
3.3	Είσοδος αλγορίθμου	31
3.3.1	Αρχείο εισόδου	31

3.3.2	Παράμετροι γραμμής εντολών	34
3.4	Εύρεση συνδεδεμένων περιοχών	35
3.5	Εύρεση ελάχιστης απόστασης μεταξύ μη συνδεδεμένων περιοχών στο τρισδιάστατο επίπεδο	36
3.6	Διαμοιρασμός της συνολικής περιοχής σε υποπεριοχές για κάθε όχημα	37
3.7	Υπολογισμός Ελαχίστου Γεννητικού Δέντρου	39
3.7.1	Αλγόριθμος του Borůvka	40
3.7.2	Αλγόριθμος του Prim	40
3.7.3	Αλγόριθμος του Kruskal	42
3.7.4	Προσπέλαση ελαχίστου γεννητικού δέντρου	43
3.8	Υπολογισμός τελικού μονοπατιού	45
3.9	Μετρικές κώδικα	46
3.10	Σύνοψη κεφαλαίου	46
4	Πειραματικές μετρήσεις	47
4.1	Παράδειγμα εκτέλεσης σε δισδιάστατο συνθετικό περιβάλλον	47
4.2	Παράδειγμα εκτέλεσης σε τρισδιάστατο συνθετικό περιβάλλον	48
4.3	Χρόνος εκτέλεσης εργαλείου	49
4.4	Σύνοψη κεφαλαίου	51
5	Συμπεράσματα και μελλοντικές βελτιώσεις	52
5.1	Συμπεράσματα	52
5.2	Προοπτική δημιουργίας επιχείρησης	53
5.2.1	Μοντέλο επιχείρησης	54
5.2.2	Ανάλυση SWOT	54
5.3	Μελλοντικές επεκτάσεις	55
5.3.1	Προσθήκη επιπλέον περιορισμών	55
5.3.2	Βελτίωση της απόδοσης του αλγορίθμου	57
5.3.3	Ανάπτυξη διαδικτυακής εφαρμογής	58
5.3.4	Ανάπτυξη Διεπαφής Προγραμματισμού Εφαρμογών (API)	58
	Παραρτήματα	60
	Α' Εγκατάσταση και χρήση εργαλείου	61

Κατάλογος σχημάτων

2.1	Παράδειγμα εντολών C#	19
2.2	Παράδειγμα ενός δέντρου.	21
2.3	Ένα γράφημα με τρία συνδεδεμένα συστατικά.	22
2.4	Παράδειγμα ενός γράφου και του Ελαχίστου Γεννητικού του Δέντρου	23
2.5	Το ολοκληρωμένο αναπτυξιακό περιβάλλον Visual Studio.	25
2.6	Ο επεξεργαστής πηγαίου κώδικα Visual Studio Code.	26
2.7	Ο επεξεργαστής πηγαίου κώδικα Notepad++.	27
3.1	Τα 4 στάδια του αλγορίθμου.	29
3.2	Παράδειγμα δισδιάστατου και τρισδιάστατου χώρου με εμπόδια . . .	30
3.3	Σταθμός εκπομπής σε δισδιάστατο χώρο.	31
3.4	Περιβάλλον δισδιάστατου χώρου 3x3.	32
3.5	Εύρεση των μη δύο μη συνδεδεμένων περιοχών.	36
3.6	Παράδειγμα διαμοιρασμού της συνολικής περιοχής ενός οχήματος χρη- σιμοποιώντας τον αλγόριθμο DARP	40
3.7	Διαμοιρασμός της συνολικής περιοχής σε κάθε όχημα.	41
3.8	Παράδειγμα υπολογισμού του Ελαχίστου Γεννητικού Δέντρου χρησι- μοποιώντας τον αλγόριθμο Kruskal	43
3.9	Βηματική εκτέλεση του αλγορίθμου Kruskal.	43
3.10	Προσπέλαση Ελαχίστου Γεννητικού Δέντρου	44
4.1	Παράδειγμα εκτέλεσης σε δισδιάστατο περιβάλλον	48
4.2	Παράδειγμα εκτέλεσης σε τρισδιάστατο περιβάλλον	50
5.1	Αποτυχία επικοινωνίας των οχημάτων με τους σταθμούς εκπομπής . .	57
5.2	Αναπαράσταση μοντέλου διαδικτυακής εφαρμογής.	58

Κατάλογος αλγορίθμων

1	Αλγόριθμος εύρεσης συνδεδεμένων στοιχείων στο επίπεδο εδάφους. . .	36
2	Αλγόριθμος εύρεσης μονοπατιού μεταξύ δύο σημείων στον τρισδιάστατο χώρο.	37
3	Αλγόριθμος του Borůvka.	41
4	Αλγόριθμος του Prim.	41
5	Αλγόριθμος του Kruskal.	42
6	Αλγόριθμος προσπέλασης ΕΓΔ για τη δημιουργία αρχικού μονοπατιού κάλυψης.	44

Κατάλογος πινάκων

3.1	Μετρικές αρχείων κώδικα.	46
5.1	SWOT ανάλυση για σύσταση επιχείρησης	55

Κατάλογος απεικονίσεων

3.1	Αρχείο εισόδου που περιγράφει το περιβάλλον του σχήματος 3.4 . . .	33
3.2	Συνάρτηση <code>ReadArrayFromFile</code>	33
4.1	Αρχείο εισόδου για το παράδειγμα εκτέλεσης δισδιάστατου χώρου . .	47
4.2	Εντολή εκτέλεσης του εργαλείου για το παράδειγμα 4.1	47
4.3	Αρχείο εισόδου για το παράδειγμα εκτέλεσης τρισδιάστατου χώρου .	49
4.4	Εντολή εκτέλεσης του εργαλείου για το παράδειγμα 4.2	49

Κατάλογος εξισώσεων

3.1	Ευκλείδεια απόσταση	38
3.2	Πίνακας αντιστοίχισης	38
3.3	Υπολογισμός της περιοχής του κάθε οχήματος	38
3.4	Πλήθος εκχωρημένων χωρορίxel ανά όχημα	38
3.5	Υπολογισμός διαθέσιμης ενέργειας	45
3.6	Υπολογισμός διαθέσιμης χωρητικότητας δεδομένων	45

Κεφάλαιο 1

Εισαγωγή

Σε αυτό του κεφάλαιο παρουσιάζεται το αντικείμενο που πραγματεύεται η διπλωματική εργασία καθώς και οι λόγοι και τα κίνητρα που οδήγησαν στην ανάληψη της. Επιπλέον, αναφέρονται παρόμοιες υλοποιήσεις άλλων ερευνητών και τέλος, γίνεται μια συνοπτική παρουσίαση της δομής της εργασίας.

1.1 Ορισμός του προβλήματος

Ζούμε αναμφίβολα στην εποχή των ραγδαίων επιστημονικών εξελίξεων και μεγάλων τεχνολογικών αλλαγών [31]. Οι εξελίξεις στην τεχνολογία ήταν πάντα ένας καθοριστικός παράγοντας που σηματοδοτούσε σημαντικές αλλαγές στην πορεία της ανθρωπότητας [30, 13]. Τα τελευταία χρόνια, ο ρυθμός εξέλιξης της τεχνολογίας επιταχύνεται θεαματικά, και ως αποτέλεσμα αυτού, η ρομποτική, τα ενσωματωμένα συστήματα και τα αυτόνομα οχήματα χρησιμοποιούνται ολοένα και περισσότερο στη ζωή του ανθρώπου.

Τα τελευταία χρόνια παρατηρείται η συνεχής ανάπτυξη και εξέλιξη των μη επανδρωμένων αεροχημάτων, γνωστά και ως drones [37, 25]. Τα drones μπαίνουν ολοένα πιο δυναμικά στη ζωή μας και βρίσκουν εφαρμογή σε πολλούς τομείς, από τις στρατιωτικές επιχειρήσεις και την ασφάλεια [7, 17], μέχρι τις αγροτικές καλλιέργειες [4] και τις τηλεοπτικές παραγωγές [16]. Έτσι, διαδικασίες όπως η φύλαξη χώρων, οι τοπογραφικές αποτυπώσεις και η καταγραφή αποτελεσμάτων από φυσικές ή μη καταστροφές βελτιώνονται σημαντικά και μειώνεται το κόστος.

Η ραγδαία εξέλιξη της τεχνολογίας σε ότι αφορά τα μη επανδρωμένα εναέρια οχήματα επιτρέπει αφενός να παρέχουν υψηλής ανάλυσης γεωγραφικά δεδομένα και από τη άλλη να μπορούν να λειτουργήσουν εξ αποστάσεως [20], επιτρέπο-

ντας στους ερευνητές να προσεγγίζουν απομακρυσμένες και δύσβατες περιοχές [41, 24, 46]. Τα μη επανδρωμένα αεροχήματα γίνονται σταδιακά εργαλεία, που δίνουν λύσεις σε εφαρμογές πολλών επιστημονικών πεδίων και είναι ικανά να διαδραματίσουν σημαντικό ρόλο στην γρήγορη λήψη γεωγραφικών πληροφοριών μεγάλης ακρίβειας με ευρύ πεδίο εφαρμογών.

Στην εποχή της ταχείας ανάπτυξης της τεχνολογίας δεν θα μπορούσε η γεωργία να μην ακολουθήσει και αυτή την τροχιά της ανάπτυξης και εξέλιξης [4, 38, 34, 42]. Η ύπαρξη τόσο μεγάλων γεωργικών μηχανημάτων όσο και μικρότερων, πιο ευέλικτων και όσο το δυνατόν αυτόνομων είναι η τάση που επικρατεί στην σύγχρονη γεωργία. Η χρήση των drones στην αγροτική παραγωγή περιγράφεται ως “γεωργία της ακριβείας“, η οποία δεν περιορίζεται μόνο σ’ αυτά αλλά είναι ένα σύστημα διαχείρισης συνόλου της γεωργικής εκμετάλλευσης με τη χρήση της πληροφορικής, του δορυφορικού εντοπισμού θέσης και των δεδομένων τηλεπισκόπησης [48, 21]. Οι τεχνολογίες αυτές έχουν ως στόχο την βελτίωση της απόδοσης της παραγωγής και παράλληλα τη μείωση των περιβαλλοντικών επιπτώσεων. Η θεωρία της ‘γεωργίας της ακριβείας’ έδωσε το έναυσμα για την προώθηση τεχνολογιών και την χρήση αυτόματων συστημάτων από τους γεωργούς, που οδηγούν στην μείωση του κόστους εργασίας, του χρόνου και χρήματος, των γεωργικών υπολειμμάτων και της ρύπανσης του περιβάλλοντος, γεγονός που απασχολεί εντονότατα την σημερινή κοινωνία και γενικότερα ολόκληρο τον πλανήτη [11, 35].

Η τεχνολογία των drones μπορεί να γίνει σύμμαχος του παραγωγού και να τον βοηθήσει να ελέγξει και να διαχειριστεί καλύτερα και με περισσότερη ακρίβεια τον ελαιώνα ή το χωράφι του και να επέμβει με ακρίβεια όπου και όποτε χρειάζεται. Τα πλεονεκτήματα ενός τέτοιου τρόπου παραγωγής είναι πολλαπλά: εξοικονόμηση και ορθολογικότερη χρήση των λιπασμάτων και των φυτοπροστατευτικών προϊόντων [44] και βελτιστοποίηση της άρδευσης [4]. Ο παραγωγός έχει μεγαλύτερο έλεγχο και την επιλογή να εφαρμόσει διαφοροποιημένες πρακτικές σε περιοχές της καλλιέργειάς του που έχουν διαφορετικές ανάγκες. Έτσι, τα μη επανδρωμένα αεροχήματα μπορούν να παρέχουν στους γεωργούς των ορεινών περιοχών της χώρας τη δυνατότητα να φεκάσουν πιο εύκολα και να ξεπεράσουν τις δυσκολίες που κρύβουν στη διαδικασία αυτή τα επικλινή εδάφη.

Εκτός βέβαια από τα εναέρια οχήματα και την γεωργία, παρατηρείται η χρήση

των αυτόνομων οχημάτων και σε υποθαλάσσιες αποστολές [32]. Ένα μη επανδρωμένο υποθαλάσσιο όχημα μπορεί να πραγματοποιεί υποθαλάσσιες παρατηρήσεις σε σημεία που τα περισσότερα ιπτάμενα ή επίγεια οχήματα δεν μπορούν. Είναι εξοπλισμένα με κάμερες και αισθητήρες υψηλής ανάλυσης, πράγμα που σημαίνει ότι μπορούν να εντοπίσουν με απίστευτη λεπτομέρεια ό,τι συμβαίνει κάτω από την επιφάνεια του νερού. Τα υποθαλάσσια οχήματα χρησιμοποιούνται για την εξερεύνηση του βυθού, για την χαρτογράφηση και συλλογή δεδομένων καθώς και σε περιπτώσεις ατυχημάτων [33].

Και ενώ τις τελευταίες δεκαετίες οι ρομποτικοί μηχανισμοί και τα αυτόνομα οχήματα έχουν μπει για τα καλά στη ζωή του ανθρώπου, [6] το πρόβλημα της δρομολόγησης αυτών χρήζει ακόμα και σήμερα λύση. Πιο συγκεκριμένα, ως δρομολόγηση ορίζεται ο σχεδιασμός μιας διαδρομής που καλύπτει όλα τα σημεία μιας προκαθορισμένης περιοχής και αποφεύγει σημεία με συγκεκριμένα χαρακτηριστικά, όπως εμπόδια και περιοχές απαγορευμένης πτήσης. Το πρόβλημα αυτό είναι γνωστό και ως βέλτιστη δρομολόγηση κάλυψης (Coverage Path Planning, CPP). Επιπλέον, είναι άμεσα συνδεδεμένο με πολλές ρομποτικές εφαρμογές, όπως υποθαλάσσια οχήματα [15], αυτόματες σκούπες ρομπότ [45], αυτόματα άρδευση [4] και εξερεύνηση του διαστήματος [19].

Το πρόβλημα αυτό, συνήθως αναφέρεται σε ένα αυτόνομο όχημα που χρησιμοποιώντας κάποιον αισθητήρα (τοποθεσίας ή περιβάλλοντος) μπορεί να εντοπίσει την θέση του στο χώρο και να καλύψει τουλάχιστον την περιοχή που το ίδιο καταλαμβάνει. Έτσι, για την καλύτερη απεικόνιση του χώρου, συνηθίζεται να χρησιμοποιείται η τεχνική διαχωρισμού του χώρου (Cell Decomposition) σε ίσα μέρη, τουλάχιστον ίσα με το μέγεθος του οχήματος [27].

Τα τελευταία χρόνια, πολλοί ερευνητές έχουν ασχοληθεί με το πρόβλημα εύρεσης μονοπατιού κάλυψης για αυτόνομα οχήματα σε προκαθορισμένο δισδιάστατο χώρο και έχουν προταθεί πολλές υλοποιήσεις. Η πιο διαδεδομένη είναι η "Κάλυψη των Ελαχίστων Δέντρων" (Spanning Tree Coverage) που εκτελείται σε γραμμικό χρόνο και εγγυάται την βέλτιστη λύση. Ως βέλτιστη λύση ορίζεται το μονοπάτι που επισκέπτεται κάθε κελί μόνο μια φορά και καλύπτει πλήρως τον ζητούμενο χώρο.

Στη παρούσα διπλωματική εργασία αναλύεται το εργαλείο που αναπτύξαμε για την εύρεση του μονοπατιού κάλυψης σε προκαθορισμένο χώρο. Σε αντίθεση με

άλλα ερευνητικά έργα, το εργαλείο μας βρίσκει το βέλτιστο μονοπάτι κάλυψης, όταν αυτό υπάρχει, σε δισδιάστατο ή τρισδιάστατο χώρο. Επιπλέον, λαμβάνονται υπόψη οι περιορισμοί των οχημάτων και του περιβάλλοντος, όπως ταχύτητα, διαθέσιμη ενέργεια, και τύπος οχήματος.

1.2 Κίνητρα και Στόχοι Υλοποίησης

Ζούμε στην Ελλάδα, μια χώρα με έντονη γεωργική δραστηριότητα όπου η σπορά, το φύτεμα, η λίπανση και το πότισμα γίνονται με ανθρώπινη χειρονακτική συμμετοχή [23]. Βέβαια, στην εποχή της ταχείας ανάπτυξης της τεχνολογίας δεν θα μπορούσε η γεωργία να μην ακολουθήσει και αυτή την τροχιά της ανάπτυξης και της εξέλιξης με στόχο να πετύχει υψηλότερες αποδόσεις στην παραγωγικότητα. Η πρόωθηση τεχνολογιών και η χρήση αυτόνομων συστημάτων από τους γεωργούς μπορεί να οδηγήσει στη μείωση του κόστους εργασίας, εξοικονόμηση χρόνου και χρήματος, στην μείωση των γεωργικών υπολειμμάτων καθώς και στην μείωση της ρύπανσης του περιβάλλοντος.

Η χρήση των ρομποτικών μηχανισμών και των αυτόνομων οχημάτων δεν περιορίζεται βέβαια μόνο στον κλάδο της γεωργίας. Αυτόνομα οχήματα χρησιμοποιούνται στην εξερεύνηση και χαρτογράφηση των υποθαλάσσιων περιοχών καθώς και στην εξερεύνηση του διαστήματος και άλλων πλανητών. Βέβαια, οι συνθήκες τόσο στην γεωργία όσο και στις υποθαλάσσιες και διαστημικές αποστολές δεν είναι ιδανικές. Έχουν ιδιαιτερότητες όπως ανώμαλα εδάφη, εδάφη με διαφορετικές ιδιότητες και χαρακτηριστικά και φυσικά πολλά εμπόδια.

Παρόλο που τα τελευταία χρόνια η χρήση των αυτόνομων οχημάτων αυξάνεται συνεχώς, παρατηρείται ταυτόχρονα πως υπάρχει μεγάλη έλλειψη από διαθέσιμο λογισμικό δρομολόγησης. Γίνεται εύκολα αντιληπτό λοιπόν πως υπάρχει μεγάλη ανάγκη να λυθεί το πρόβλημα της βέλτιστης δρομολόγησης αυτόνομων οχημάτων.

Λαμβάνοντας υπόψη τη σημασία της βέλτιστης δρομολόγησης αυτόνομων οχημάτων καθώς και την έλλειψη λογισμικού, αναπτύξαμε ένα εργαλείο βέλτιστης δρομολόγησης αυτόνομων οχημάτων. Το εργαλείο δέχεται ως είσοδο την περιοχή ενδιαφέροντος σε μορφή τρισδιάστατου πίνακα καθώς και πληροφορίες σχετικά με τα χαρακτηριστικά των οχημάτων (όπως αρχική τοποθεσία, ταχύτητα, συνολική ενέργεια και αποθηκευτικός χώρος). Σαν έξοδο, παράγει την διαδρομή που θα ακολουθήσει

το κάθε όχημά προκειμένου να καλυφθεί όλος ο ζητούμενος χώρος. Η παραγόμενη διαδρομή είναι προσαρμοσμένη στο κάθε όχημα βάσει των χαρακτηριστικών του.

1.3 Περιπτώσεις παρόμοιων ερευνητικών έργων

Τις τελευταίες δεκαετίες η δημοτικότητα των ρομποτικών μηχανισμών και των αυτόνομων οχημάτων αυξάνεται με ολοένα και μεγαλύτερο ρυθμό [8]. Ως εκ τούτου, το πρόβλημα της βέλτιστης κάλυψης χώρου έχει προσελκύσει μεγάλη προσοχή απο πολλούς ερευνητές. Υπάρχουν πολλές διαφορετικές προσεγγίσεις που προσπαθούν να λύσουν το πρόβλημα της βέλτιστης δρομολόγησης και συγκεκριμένα της δρομολόγησης κάλυψης χώρου.

Ένας αλγόριθμος κάλυψης χώρου για αυτόνομα οχήματα καθαρισμού παρουσιάζεται στο [45]. Οι ερευνητές χρησιμοποίησαν μια επέκταση της κυτταρικής αποσύνθεσης [10] σε συνδυασμό με τον αλγόριθμο Dijkstra [12, 29]. Παρόλο που ο συγκεκριμένος αλγόριθμος παράγει ένα μονοπάτι κάλυψης χώρου αποφεύγοντας τα εμπόδια, λειτουργεί σε καταστάσεις ενός μόνο οχήματος και είναι αναποτελεσματικός όσον αφορά τις απαιτήσεις μνήμης και αποθήκευσης.

Ο Gabriely κ.α. [14] πρότειναν έναν αλγόριθμο σπειροειδούς κάλυψης ελαχίστου γεννητικού δέντρου (Spanning Tree Coverage, STC), βασισμένο στην ιδέα της διαίρεσης του περιβάλλοντος σε χώρους διπλάσιους σε διαστάσεις από το μέγεθος του οχήματος (κατά προσέγγιση κυτταρική αποσύνθεση, approximate cell decomposition) [27]. Ο αλγόριθμός που παρουσιάζεται προσφέρει πλήρη κάλυψη της ζητούμενης δισδιάστατης περιοχής και αποφεύγονται τα εμπόδια. Ωστόσο, η προτεινόμενη μέθοδος είναι ικανή να χρησιμοποιεί μόνο ένα όχημα κάθε φορά. Επιπλέον, δεν λαμβάνονται υπόψη περιορισμοί όπως η ταχύτητα των οχημάτων και η διαθέσιμη ενέργεια.

Ο Καπουτσης κ.α. [22] προσπάθησαν να λύσουν το πρόβλημα της βέλτιστης κάλυψης χώρου διαιρώντας την περιοχή ενδιαφέροντος σε ίσες υποπεριοχές με βάση τον αριθμό των διαθέσιμων οχημάτων. Μετά την διαίρεση της αρχικής περιοχής σε υποπεριοχές, κάθε όχημα χρησιμοποιεί την τεχνική κάλυψης ελαχίστου γεννητικού δέντρου (STC) για να καλύψει την υποπεριοχή που ανήκει σε αυτό. Παρόλο που ο προτεινόμενος αλγόριθμός λειτουργεί σε καταστάσεις πολλαπλών οχημάτων, δεν λαμβάνονται υπόψη οι διαφορετικοί τύποι οχημάτων, η διαθέσιμη ενέργεια και η ταχύτητά τους. Επιπλέον, σημαντικό μειονέκτημα είναι το γεγονός πως δεν μπορεί

να χρησιμοποιηθεί και για την κάλυψη περιοχών σε τρισδιάστατα περιβάλλοντα.

Ο Hector κ.α. [5] πρότειναν μια μεθοδολογία για δρομολόγηση αυτόνομων εναέριων οχημάτων (drones) σε τρισδιάστατο χώρο για καταστάσεις έκτακτης ανάγκης μετά από ένα καταστροφικό γεγονός όπως μια πυρκαγιά ή ένας σεισμός. Χρησιμοποιείται η τεχνική προσεγγιστικής κυτταρικής αποσύνθεσης χώρου (approximate cell decomposition) σε εξάγωνα ίσων διαστάσεων. Μεγάλο μειονέκτημα του προτεινόμενου αλγορίθμου είναι πως λειτουργεί μόνο σε καταστάσεις εναέριων οχημάτων. Επίσης, δεν είναι δυνατή η χρήση οχημάτων με διαφορετικά χαρακτηριστικά όπως ταχύτητα.

Το έργο μας διαφέρει από τα προαναφερθέντα σε δύο βασικά σημεία: (α) λύνουμε το πρόβλημα κάλυψης χώρου πολλαπλών οχημάτων σε τρισδιάστατα περιβάλλοντα και (β) λαμβάνουμε υπόψη ειδικά χαρακτηριστικά των οχημάτων, όπως ταχύτητα και τύπο. Επιπλέον, η μεθοδολογία που παρουσιάζεται στην παρούσα διπλωματική εργασία λύνει το πρόβλημα της κάλυψης χώρου σε καταστάσεις πολλαπλών οχημάτων σε δισδιάστατα και τρισδιάστατα περιβάλλοντα χρησιμοποιώντας έναν ενιαίο αλγόριθμο. Επιπλέον, οι περιορισμοί και τα χαρακτηριστικά των οχημάτων είναι πλήρως αρθρωτά, πράγμα που σημαίνει ότι μπορούμε να επιλέξουμε κατά την εκτέλεση ποιοι περιορισμοί ισχύουν για κάθε όχημα. Δεδομένου ότι εισάγουμε πολλούς περιορισμούς στο πρόβλημα της εύρεσης μονοπατιού κάλυψης χώρου, ο χρόνος εκτέλεσης του προτεινόμενου αλγορίθμου είναι μεγαλύτερος από τον χρόνο εκτέλεσης των προαναφερθέντων αλγορίθμων.

1.4 Διάρθρωση κειμένου

Τα υπόλοιπα κεφάλαια οργανώνονται ως εξής: Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο των εννοιών που χρησιμοποιούνται. Ακολουθεί η περιγραφή του λογισμικού μέρους στο κεφάλαιο 3. Έπειτα, στο κεφάλαιο 4 δίνονται ορισμένα πειραματικά αποτελέσματα που προέκυψαν από την εκτέλεση του αλγορίθμου σε συνθετικά περιβάλλοντα. Τέλος, το κεφάλαιο 5 εξετάζει την πιθανότητα ίδρυσης επιχείρησης με βάση το εργαλείο που παρουσιάζεται στη παρούσα διπλωματική εργασία και κλείνει με συμπεράσματα και μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο που ακολουθεί παρουσιάζονται όλα τα μέσα που συντέλεσαν στην υλοποίηση της διπλωματικής εργασίας. Παρουσιάζονται οι γλώσσες προγραμματισμού, εξηγούνται οι λέξεις κλειδιά και δίνονται τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν ώστε ο αναγνώστης να είναι σε θέση να παρακολουθήσει τα επόμενα κεφάλαια.

2.1 C-Sharp

Η C-Sharp (C#) [1] είναι μια γλώσσα προγραμματισμού Ηλεκτρονικών Υπολογιστών. Δημιουργήθηκε από την Microsoft μέσα από την πλατφόρμα .NET και αργότερα αναγνωρίστηκε επισήμως από την ECMA (ECMA-334) και την ISO (ISO/IEC 2327:2006). Η C# είναι μια αντικειμενοστραφείς γλώσσα γενικού τύπου που μπορεί να χρησιμοποιηθεί σε πολλές πλατφόρμες όπως Windows, Mac OS και Linux [2]. Επιπλέον, χρησιμοποιώντας την πλατφόρμα Xamarin είναι πλέον δυνατή η χρήση της C# για ανάπτυξη εφαρμογών για κινητά τηλέφωνα (mobile applications) [3].

Συγκριτικά με τις γλώσσες προγραμματισμού C και C++, παρατηρούμε πως η C# υπερέχει σε αρκετούς τομείς όπως:

- στην αυτόματη διαχείριση απορριμάτων (garbage collector)
- δεν είναι απαραίτητη η χρήση δεικτών, ωστόσο υποστηρίζονται πλήρως
- αυτόματη διαχείριση μνήμης
- εύκολη ανάπτυξη πολυνηματικών εφαρμογών (multithreading)

- ασφάλεια τύπου (type safety)

Συγκριτικά με την γλώσσα προγραμματισμού Java, η C# υπερέρχει σε τομείς όπως:

- καλύτερη διαχείριση γεγονότων χρησιμοποιώντας delegates
- υπερφόρτωση τελεστών
- πλήρης υποστήριξη δεικτών, ωστόσο δεν είναι απαραίτητη η χρήση τους
- καλύτερη υποστήριξη απαρίθμησης, χρησιμοποιώντας την δήλωση yield

Για τους λόγους αυτούς επιλέχθηκε η C# έναντι των άλλων διαθέσιμων γλωσσών προγραμματισμού. Στο σχήμα 2.1 εμφανίζεται ένα απλό παράδειγμα εντολών της γλώσσας προγραμματισμού C#.

```
1  using System;
2
3  namespace HelloWorld
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello world!");
10         }
11     }
12 }
```

Σχήμα 2.1: Παράδειγμα εντολών C#

2.2 Αλγόριθμος αναζήτησης

Στην επιστήμη των υπολογιστών, ένας αλγόριθμος αναζήτησης είναι ένας αλγόριθμος για την εύρεση ενός αντικειμένου με συγκεκριμένες ιδιότητες μεταξύ μιας συλλογής αντικειμένων. Τα αντικείμενα μπορεί να βρίσκονται αποθηκευμένα ατομικά ως δεδομένα σε μια δομή δεδομένων ή να είναι στοιχεία ενός χώρου αναζήτησης προσδιορισμένου από μια μαθηματική παράσταση ή διαδικασία, όπως οι ρίζες μιας εξίσωσης με ακέραιες μεταβλητές, ή ένας συνδυασμός των δύο, όπως οι κύκλοι Hamilton ενός γράφου.

2.3 Αλγόριθμος Coverage Path Planning (CPP)

Αλγόριθμος CPP ή αλγόριθμος εύρεσης μονοπατιού κάλυψης ονομάζεται ένας αλγόριθμος που καλείται να βρεί ένα μονοπάτι, το οποίο διέρχεται από όλα τα προσβάσιμα σημεία ενώ ταυτόχρονα αποφεύγει τα εμπόδια. Οι αλγόριθμοι CPP χωρίζονται σε δύο μεγάλες κατηγορίες [9]:

- Online αλγόριθμοι

Στους online αλγορίθμους, η μορφολογία και οι ιδιαιτερότητές του περιβάλλοντος δεν είναι γνωστά εξ αρχής. Έτσι, χρησιμοποιώντας αισθητήρες, το όχημα ή ο ρομποτικός μηχανισμός που εκτελεί τον εκάστοτε αλγόριθμο, λαμβάνει πληροφορίες σχετικά με τον χώρο που το περιβάλλει και στη συνέχεια προσπαθεί να πλοηγηθεί μέσα σε αυτό.

- Offline αλγόριθμοι

Σε αντίθεση με τους online αλγορίθμους, οι offline αλγόριθμοι έχουν πλήρη γνώση εξ αρχής για όλο το περιβάλλον. Σε αυτήν την κατηγορία εμπίπτουν μεθοδολογίες όπως γενετικοί αλγόριθμοι, νευρωνικά δίκτυα, κυτταρική αποσύνθεση (cellular decomposition), γεννητικά δέντρα (spanning trees), μονοπάτια σπειροειδούς πλήρωσης (spiral filling paths) και η μέθοδος αποικίας μυρμηγκιού (ant colony method) [43]

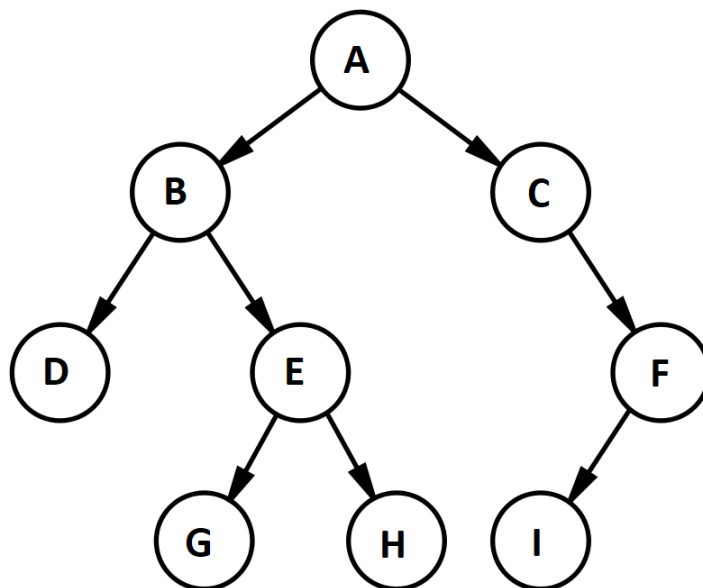
2.4 Γράφημα

Ένα γράφημα $G = (V, E)$ ορίζεται ως μια μαθηματική δομή η οποία αποτελείται από δύο σύνολα: το σύνολο των ακμών E και το μη κενό σύνολο των κορυφών V . Κάθε ακμή του συνόλου E αποτελείται από ένα ή δύο στοιχεία του συνόλου των κορυφών V , τα οποία ονομάζονται άκρα της ακμής, και αναπαριστά τη σύνδεση μεταξύ των δύο κορυφών ή ότι η κορυφή συνδέεται με τον εαυτό της. Η ύπαρξη μιας ακμής υποδηλώνει εννοιολογική ή φυσική σχέση μεταξύ των κορυφών. Γραφήματα στα οποία ένα ζεύγος κορυφών ενώνεται μόνο με μια ακμή ονομάζονται απλά γραφήματα.

Στα γραφήματα που οι ακμές αποτελούνται από απλές γραμμές χωρίς κατευθύνσεις και δεν περιέχονται βρόγχοι, δηλαδή ακμές με αρχή και τέλος την ίδια κορυφή, το γράφημα ονομάζεται μη κατευθυνόμενο. Ωστόσο, σε ορισμένες σχέσεις η ύπαρξη κατευθύνσεων από μια κορυφή σε μια άλλη είναι απαραίτητη. Σε αυτές τις περιπτώσεις η ιδιότητα των κατευθύνσεων προστίθεται στις ακμές και το γράφημα ονομάζεται κατευθυνόμενο. Οι ακμές επίσης μπορούν να συσχετιστούν με αριθμούς οι οποίοι καλούνται βάρη. Τα γραφήματα που περιέχουν τέτοιες ακμές ανήκουν στην κατηγορία των βεβαρημένων γραφημάτων.

2.5 Δενδρική δομή

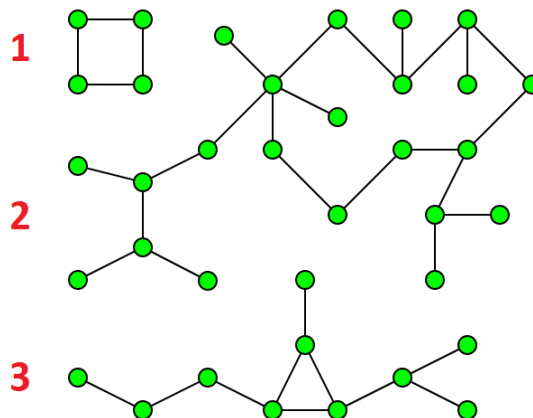
Στη θεωρία γραφημάτων, ως δενδρική δομή ή δέντρο ορίζεται ένα συνεκτικό και μη κατευθυνόμενο γράφημα το οποίο δεν περιέχει κύκλους. Η δενδρική δομή είναι ένας τρόπος να παραστήσουμε γραφικά την ιεραρχία μιας δομής. Είναι η δομή δεδομένων που αποτελείται από ένα σύνολο κόμβων που συνδέονται με ακμές. Σε κάθε μία πεπερασμένη δενδρική δομή υπάρχει ο ανώτατος αρχικός κόμβος από τον οποίο ξεκινάει η διακλάδωση, και ονομάζεται ρίζα του δέντρου. Οι κόμβοι που δεν διακλαδώνονται περαιτέρω ονομάζονται φύλλα του δέντρου. Οι κόμβοι μεταξύ τους έχουν σχέσεις γονιού - παιδιού. Κάθε παιδί έχει μόνον ένα γονιό. Στο σχήμα 2.2 εμφανίζεται ένα παράδειγμα ενός δέντρου.



Σχήμα 2.2: Παράδειγμα ενός δέντρου. Ο κόμβος A αποτελεί την ρίζα του δέντρου.

2.6 Συνδεδεμένο στοιχείο (Connected Component)

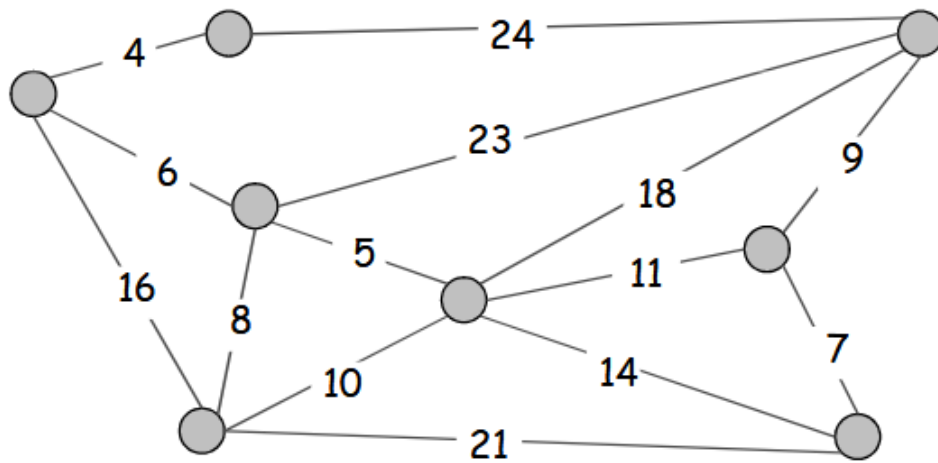
Στη θεωρία των γράφων, ένα συνδεδεμένο στοιχείο ενός μη κατευθυνόμενου γραφήματος είναι ένα υπογράφημα στο οποίο οποιεσδήποτε δύο κορυφές συνδέονται μεταξύ τους με μονοπάτια. Για παράδειγμα, το γράφημα που παρουσιάζεται στο σχήμα 2.3 έχει τρία συνδεδεμένα στοιχεία. Ένα γράφημα που είναι συνδεδεμένο έχει ακριβώς ένα συνδεδεμένο στοιχείο, το οποίο αποτελείται από ολόκληρο το γράφημα.



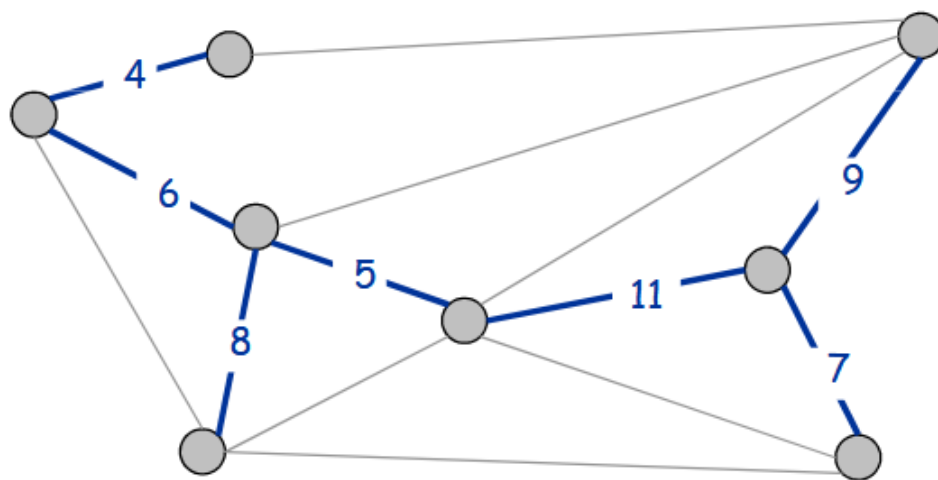
Σχήμα 2.3: Ένα γράφημα με τρία συνδεδεμένα συστατικά.

2.7 Ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree)

Το ελάχιστο γεννητικό δέντρο (Minimum Spanning Tree, MST) ή γεννητικό δέντρο ελαχίστου κόστους είναι ένα υποσύνολο των άκρων ενός συνδεδεμένου μη κατευθυνόμενου βεβαρημένου γραφήματος, που συνδέει όλες τις κορυφές, χωρίς κύκλους και με το ελάχιστο συνολικό βάρος ακμής. Είναι ένα δέντρο δηλαδή που το άθροισμά των βαρών των ακμών είναι όσο το δυνατόν μικρότερο. Αν τα βάρη των ακμών του γράφου είναι διάφορα μεταξύ τους, τότε το ελάχιστο γεννητικό δέντρο του είναι μοναδικό. Διαφορετικά, αν υπάρχουν ακμές με το ίδιο βάρος, τότε μπορεί να υπάρχουν δύο ή και παραπάνω ελάχιστα γεννητικά δέντρα. Στο σχήμα 2.4 φαίνεται σχηματικά το ελάχιστο γεννητικό δέντρο ενός γράφου.



(α)



(β)

Σχήμα 2.4: Στο σχήμα (α) βλέπουμε έναν γράφο με βάρη στις ακμές του. Το ελάχιστο γεννητικό δέντρο (MST) του γράφου φαίνεται στο σχήμα (β).

2.8 Νήματα (threads)

Νήμα (thread) ονομάζεται μια ανεξάρτητη ακολουθία ελέγχων και εντολών εντός προγράμματος. Πολλές φορές, η έννοια του νήματος συγχέεται με αυτή της διεργασίας, καθώς τα νήματα αναφέρονται συχνά σαν ελαφριές διεργασίες. Η ειδοποιός διαφορά ανάμεσα τους είναι ότι τα νήματα μπορούν να μοιράζονται τον ίδιο χώρο διευθύνσεων ενώ αντίθετα οι διεργασίες συνήθως όχι. Έτσι, ενώ δύο νήματα ενός προγράμματος έχουν πρόσβαση σε κάποιο αντικείμενο αυτού, δύο διεργασίες βλέπουν δυο διαφορετικά αντίγραφα του ίδιου αντικειμένου. Κάθε διεργασία μπορεί να διαθέτει περισσότερα από ένα νήματα, τα οποία περιέχουν τις εντολές προς την εκτελούμενη διεργασία καθώς και το εννοιολογικό της πλαίσιο, δηλαδή τις τιμές των μεταβλητών την κάθε χρονική στιγμή.

Σε έναν μονοπύρηνο επεξεργαστή, η πολυνημάτωση (multithreading) πραγματοποιείται με την μέθοδο της πολυπλεξίας με διαίρεση του χρόνου. Πιο συγκεκριμένα, ο επεξεργαστής μεταπηδάει μεταξύ των διάφορων νημάτων ανα πολύ τακτά χρονικά διαστήματα, έτσι ώστε ο χρήστης έχει την εντύπωση ότι τα νήματα εκτελούνται την ίδια στιγμή. Μόνο σε έναν πολυπύρηνο επεξεργαστή τα νήματα εκτελούνται πραγματικά ταυτόχρονα και κάθε πυρήνας εκτελεί ένα συγκεκριμένο νήμα ή εργασία.

Τα πλεονεκτήματα του πολυνηματικού προγραμματισμού περιλαμβάνουν την πιο αποτελεσματική χρήση των πόρων καθώς και την καλύτερη απόκριση των προγραμμάτων. Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκε πολυνηματικός προγραμματισμός για την βελτίωση της απόδοσης του προγράμματος και μείωση του χρόνου εκτέλεσης ορισμένων αλγορίθμων.

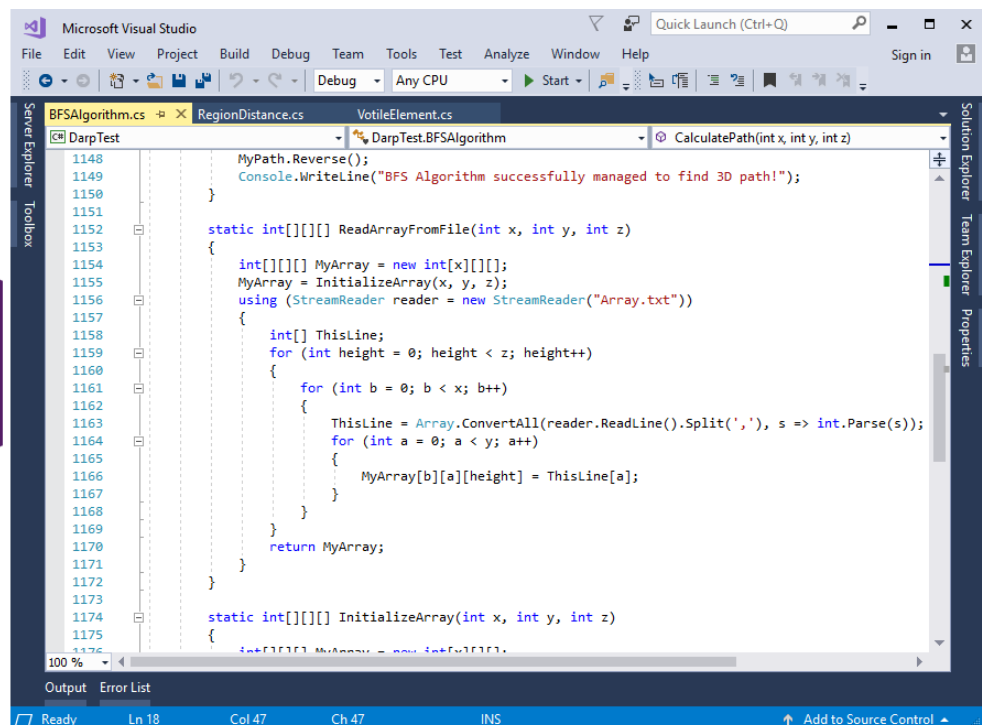
2.9 Εργαλεία που χρησιμοποιήθηκαν

2.9.1 Visual Studio

Το Visual Studio είναι το επίσημο ολοκληρωμένο αναπτυξιακό περιβάλλον (Integrated Development Environment) της Microsoft για την ανάπτυξη προγραμμάτων, ιστοσελίδων και εφαρμογών κινητών τηλεφώνων. Το Visual Studio χρησιμοποιεί τις πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως Windows API, Windows Forms

και Microsoft Silverlight ενώ μπορεί να παράγει κώδικα μηχανής (native code) καθώς και διαχειριζόμενο κώδικα (managed code).

Το Visual Studio περιλαμβάνει επεξεργαστή κειμένου, αποσφαλματωτή (debugger), εργαλείο για ανάπτυξη εφαρμογών με γραφικό περιβάλλον (GUI applications) καθώς και πρόγραμμα διαχείρισης εκδόσεων (source control). Επιπλέον, υποστηρίζει συνολικά 36 γλώσσες προγραμματισμού, όπως η C, C#, C++, Python, HTML και CSS. Η βασική έκδοση του Visual Studio (Community Edition) διατίθεται δωρεάν. Στην παρούσα διπλωματική χρησιμοποιήθηκε η έκδοση 2017 Community Edition. Στο σχήμα 2.5 απεικονίζεται το εικονίδιο καθώς και ένα στιγμιότυπο της οθόνης του Visual Studio.

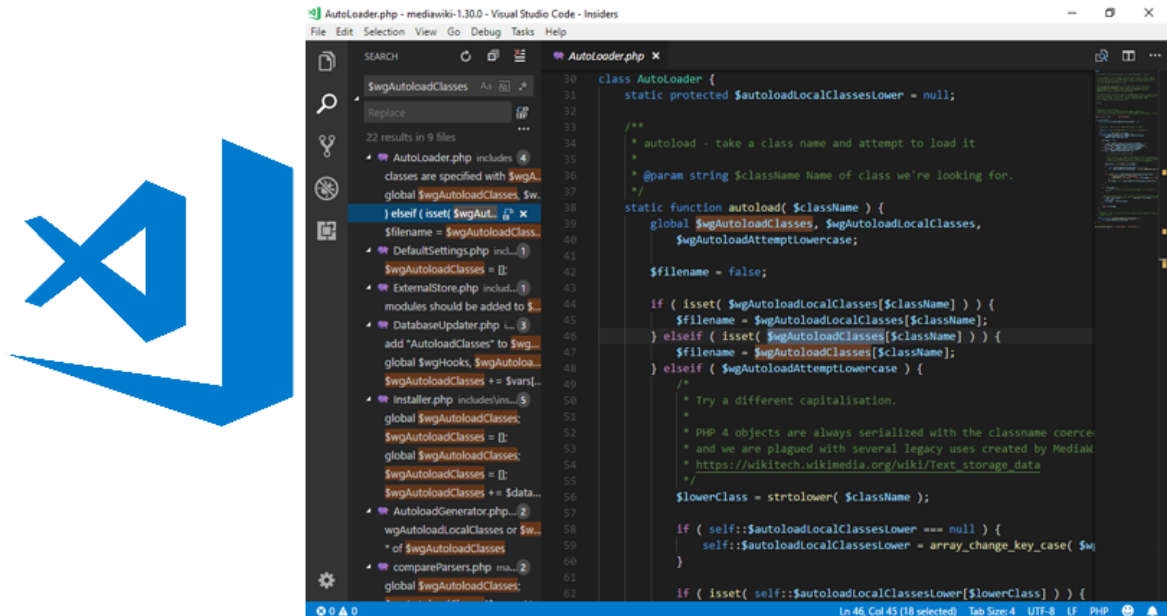


Σχήμα 2.5: Το ολοκληρωμένο αναπτυξιακό περιβάλλον Visual Studio.

2.9.2 Visual Studio Code

Το Visual Studio Code είναι ένας δωρεάν επεξεργαστής πηγαίου κώδικα από την Microsoft. Υποστηρίζει πλατφόρμες όπως Microsoft Windows, Mac OS και Linux. Περιλαμβάνει αποσφαλματωτή (debugger), ενσωματωμένο διαχειριστή εκδόσεων (source control), υπογραμμισμός σύνταξης και συντομεύσεις πλήκτρων. Είναι λογισμικό ανοιχτού κώδικα και μπορεί να παραμετροποιηθεί σε μεγάλο βαθμό ανάλογα

με τις προτιμήσεις του χρήστη χρησιμοποιώντας επιπρόσθετα (plugins). Στο σχήμα 2.6 απεικονίζεται ένα στιγμιότυπο της οθόνης του Visual Studio Code.



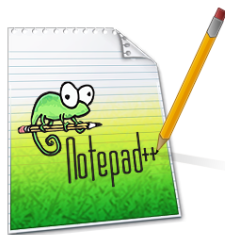
Σχήμα 2.6: Ο επεξεργαστής πηγαίου κώδικα Visual Studio Code.

2.9.3 Notepad++

Το Notepad++ είναι ένας δωρεάν επεξεργαστής πηγαίου κώδικα για το λειτουργικό σύστημα Microsoft Windows. Υποστηρίζει ταυτόχρονη επεξεργασία πολλαπλών αρχείων χρησιμοποιώντας καρτέλες. Το Notepad++ είναι λογισμικό ανοιχτού κώδικα και διέπεται από την άδεια GPL (Γενική Δημόσια Άδεια). Χρησιμοποιήθηκε στη συγγραφή του κώδικα της εφαρμογής. Στο σχήμα 2.7 απεικονίζεται το εικονίδιο και ένα στιγμιότυπο της οθόνης του Notepad++.

2.9.4 Git

Για την εύκολη διαχείριση του έργου, χρησιμοποιήθηκε το git. Πρόκειται για ένα σύστημα ελέγχου εκδόσεων με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Linus Torvalds για την ανάπτυξη του πυρήνα Linux το 2005. Σήμερα, το git αποτελεί το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού και πάνω του στηρίζονται υπηρεσίες όπως το gitlab και github.



```
ReadArrayFromFile.cs - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Window ?
ReadArrayFromFile.cs
1 using System;
2 using System.Collections.Generic;
3 using System.IO;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace CPP3D
9 {
10     public class FileIQ
11     {
12         static int[][][] ReadArrayFromFile(int x, int y, int z)
13         {
14             int[][][] MyArray = new int[x][][];
15             MyArray = InitializeArray(x, y, z);
16             using (StreamReader reader = new StreamReader("Array.txt"))
17             {
18                 int[] ThisLine;
19                 for (int height = 0; height < z; height++)
20                 {
21                     for (int b = 0; b < y; b++)
22                     {
23                         ThisLine = Array.ConvertAll(reader.ReadLine().Split(','), s => int.Parse(s));
24                         for (int a = 0; a < x; a++)
25                         {
26                             MyArray[b][a][height] = ThisLine[a];
27                         }
28                     }
29                 }
30                 return MyArray;
31             }
32         }
33     }
34 }
35
C# source file length: 1,017 lines: 35 Ln: 10 Col: 24 Sel: 0|0 Windows (CR LF) UTF-8 INS
```

Σχήμα 2.7: Ο επεξεργαστής πηγαίου κώδικα Notepad++.

2.10 Σύνοψη Κεφαλαίου

Σε αυτό το κεφάλαιο έγινε αναφορά σε όλα τα μέσα που συντέλεσαν στην υλοποίηση της παρούσας διπλωματικής εργασίας. Παρουσιάστηκαν οι γλώσσες προγραμματισμού, ερμηνεύτηκαν οι λέξεις κλειδιά και δόθηκαν τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν. Στο κεφάλαιο που ακολουθεί, περιγράφεται με λεπτομέρεια ο διαχωρισμός του λογισμικού μέρους του εργαλείου στα επιμέρους τμήματα που το αποτελούν και επιπλέον δίνεται η ανάλυση για κάθε του τμήμα.

Κεφάλαιο 3

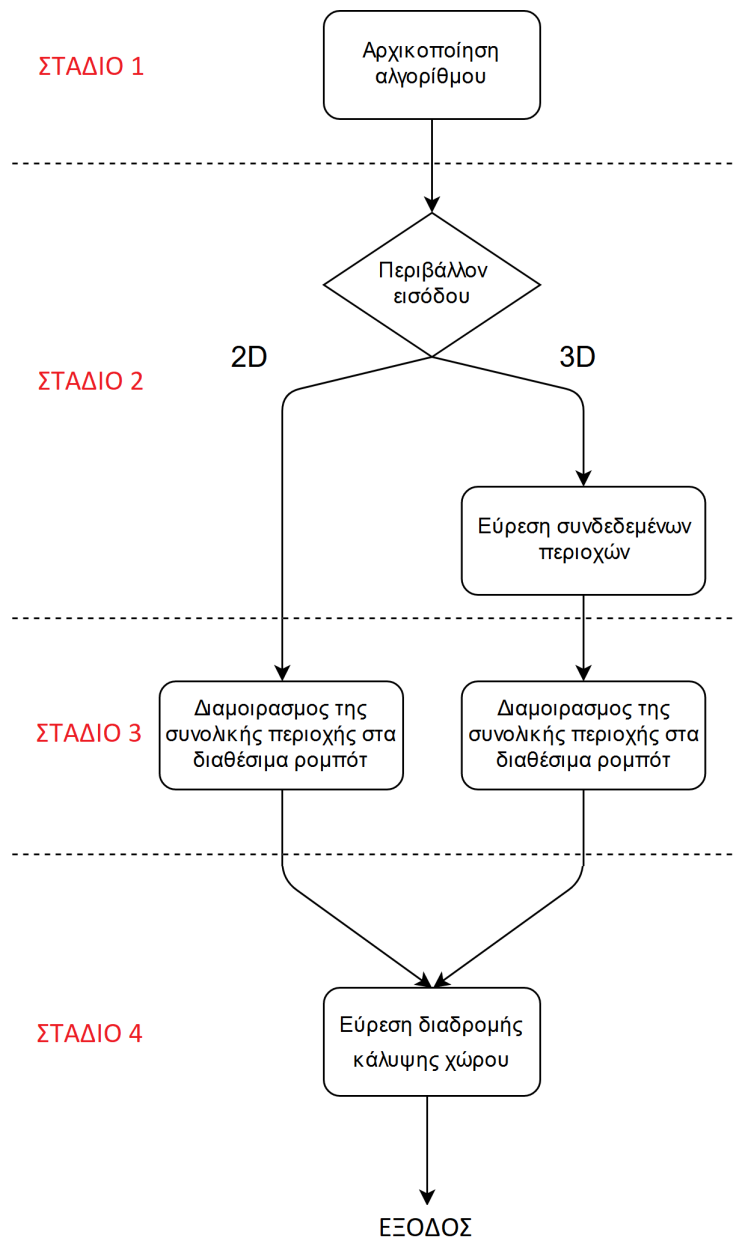
Υλοποίηση του λογισμικού μέρους

Στο κεφάλαιο αυτό παρουσιάζεται το λογισμικό που αναπτύχθηκε και τα στάδια που περιλαμβάνει. Επιπλέον, δίνονται και ερμηνεύονται οι αλγόριθμοι που χρησιμοποιήθηκαν.

3.1 Γενική επισκόπηση του λογισμικού

Ο προτεινόμενος αλγόριθμος αποτελείται από τέσσερα κύρια στάδια (σχήμα 3.1). Το πρώτο στάδιο ασχολείται με την αρχικοποίηση του αλγορίθμου. Η είσοδος του αλγορίθμου περιλαμβάνει: το μέγεθος και την μορφολογία του περιβάλλοντος, τις θέσεις στις οποίες βρίσκονται εμπόδια καθώς και τις προδιαγραφές των οχημάτων (όπως τύπος, ταχύτητα και αρχικές θέσεις). Αυτά τα δεδομένα μπορούν να διαβαστούν απευθείας από ένα αρχείο κειμένου (text file). Ο αλγόριθμος αποθηκεύει στη συνέχεια όλα τα δεδομένα στη μνήμη για ταχύτερη πρόσβαση κατά την εκτέλεση. Το δεύτερο στάδιο ασχολείται με την εύρεση συνδεδεμένων περιοχών. Αρχικά, εντοπίζονται (αν υπάρχουν) οι περιοχές που δεν είναι προσβάσιμες από το έδαφος. Έπειτα, ο αλγόριθμος βρίσκει την βέλτιστη διαδρομή που ενώνει τις συγκεκριμένες περιοχές. Στην περίπτωση που το περιβάλλον εισόδου αποτελείται από διδιάστατο χώρο, το βήμα αυτό παραλείπεται. Στο τρίτο στάδιο, η συνολική περιοχή χωρίζεται σε πολλαπλές υποπεριοχές, μια για κάθε όχημα. Αυτό το στάδιο εξάγει πίνακες δύο ή τριών διαστάσεων που περιέχουν πληροφορίες σχετικά με την εκχώρηση περιοχής στα οχήματα. Στην περίπτωση ενός μόνο οχήματος, ο αλγόριθμος εκχωρεί απευθείας όλη την συνολική αρχική περιοχή στο όχημα χωρίς περαιτέρω υπολογισμούς. Στο τέταρτο και τελευταίο στάδιο, για κάθε υποπεριοχή, υπολογίζεται η διαδρομή κάλυψης. Η έξοδος του αλγορίθμου περιλαμβάνει ένα αρ-

χείο εξόδου για κάθε όχημα που περιγράφει τη διαδρομή κάλυψης.

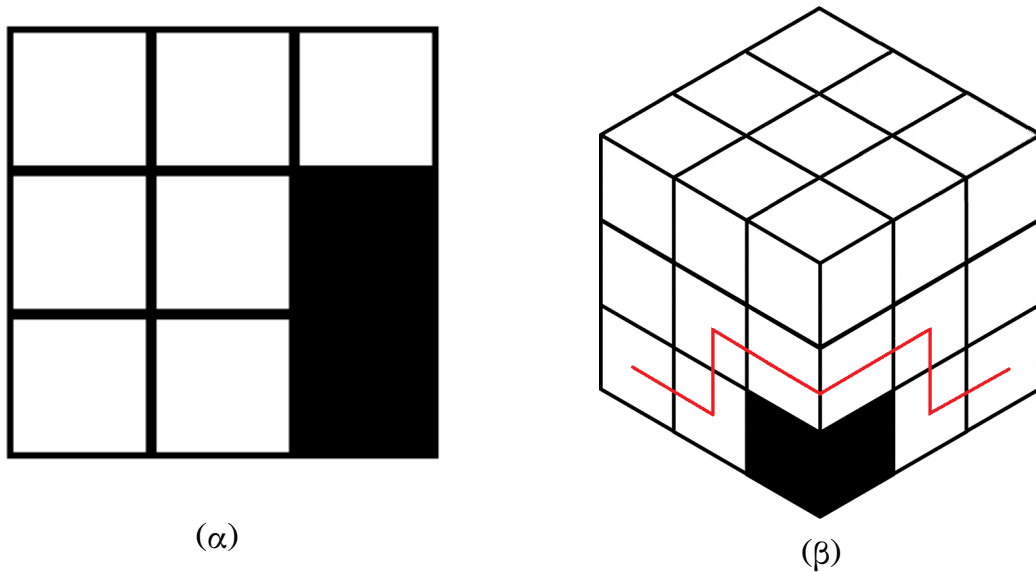


Σχήμα 3.1: Τα 4 στάδια του αλγορίθμου.

3.2 Περιγραφή του περιβάλλοντος

Το περιβάλλον στο οποίο ο αλγόριθμος καλείται να υπολογίσει το μονοπάτι πλήρους κάλυψης μπορεί να είναι ένας δισδιάστατος ή τρισδιάστατος Ευκλείδειος χώρος. Ο χώρος αυτός αποτελείται από πολλαπλές υποπεριοχές που ονομάζουμε χωροpixel (σε αντίθεση με τα κελιά που υπάρχουν μόνο στα δισδιάστατα περιβάλλοντα). Τα χωροpixel φέρουν τρεις διαστάσεις και αποτελούν διευθυνσιοδοτούμενες

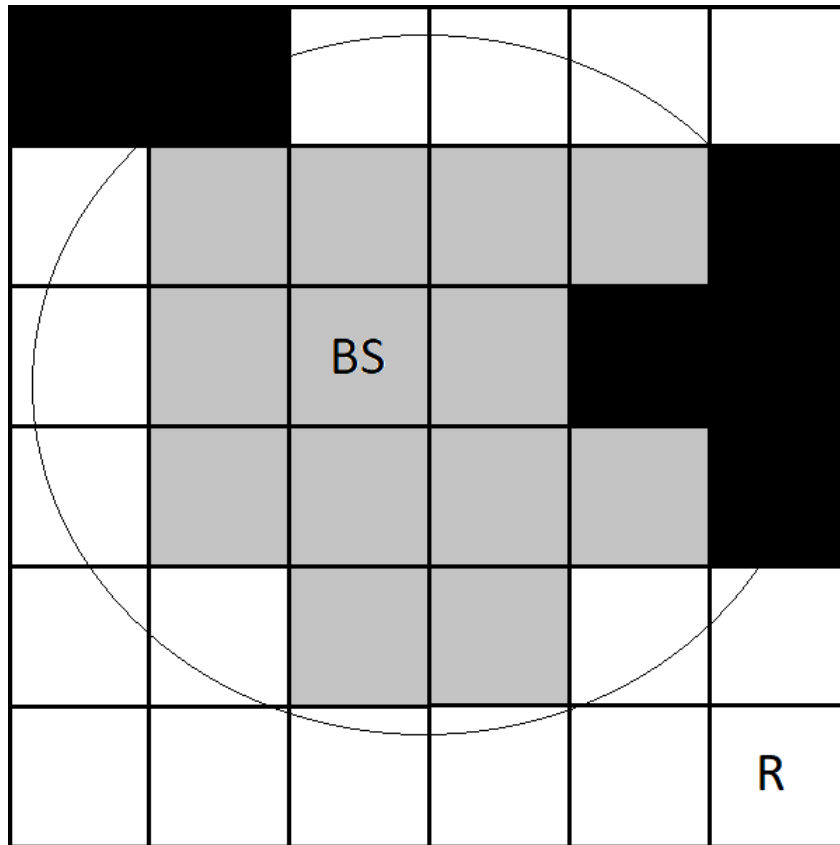
περιοχές του χώρου. Κάθε χωροpixel βρίσκεται σε ένα σημείο (X, Y, Z) . Το ελάχιστο μέγεθος που μπορεί να έχει ένα χωροpixel είναι διπλάσιο του μεγέθους του οχήματος. Επιπλέον, όλα τα χωροpixel έχουν τις ίδιες διαστάσεις. Κάθε χωροpixel μπορεί να υποδηλώνει ελεύθερο χώρο (προσβάσιμη περιοχή) ή εμπόδιο (μη προσβάσιμη περιοχή). Στα τρισδιάστατα περιβάλλοντα, τα εμπόδια μπορούν να βρίσκονται σε συγκεκριμένα ύψη, εμποδίζοντας για παράδειγμα τα οχήματα εδάφους αλλά επιτρέποντας την προσπέλαση από πάνω στα ιπτάμενα οχήματα (σχήμα 3.2).



Σχήμα 3.2: Ένας διδιάστατος (α) και ένας τρισδιάστατος (β) χώρος. Τα χωροpixel που είναι προσβάσιμα από τα οχήματα σημειώνονται με λευκό χρώμα και τα εμπόδια σημειώνονται με μαύρο χρώμα. Στο (β) βλέπουμε ένα όχημα να προσπερνάει ένα εμπόδιο πετώντας από πάνω του.

Ένα επιπλέον χαρακτηριστικό του περιβάλλοντος είναι οι πηγές ενέργειας και οι σταθμοί εκπομπής. Οι πηγές ενέργειας βρίσκονται σε ορισμένα προκαθορισμένα σημεία στον χώρο. Τα οχήματα μπορούν να επισκέπτονται τα σημεία αυτά και να φορτίζουν τις μπαταρίες τους, επεκτείνοντας έτσι την αυτονομία τους. Στα τρισδιάστατα περιβάλλοντα, οι πηγές ενέργειας μπορούν να βρίσκονται σε οποιοδήποτε ύψος του χώρου.

Οι σταθμοί εκπομπής είναι ουσιαστικά κεραιές που επιτρέπουν την μετάδοση πληροφοριών από και προς τα οχήματα. Έχουν κυκλική εμβέλεια και τα οχήματα μπορούν να επισκέπτονται χωροpixel εντός της εμβέλειας ανά τακτά χρονικά διαστήματα ώστε να μεταφορτώνουν τα δεδομένα που έχουν συλλέξει.



Σχήμα 3.3: Σταθμός εκπομπής σε διδιάστατο χώρο. Τα γκρι χωρορίxel υποδηλώνουν περιοχές εντός της εμβέλειας του σταθμού εκπομπής. Τα μαύρα χωρορίxel υποδηλώνουν εμπόδια. Τα λευκά χωρορίxel υποδηλώνουν ελεύθερη και προσβάσιμη περιοχή από τα οχήματα.

3.3 Είσοδος αλγορίθμου

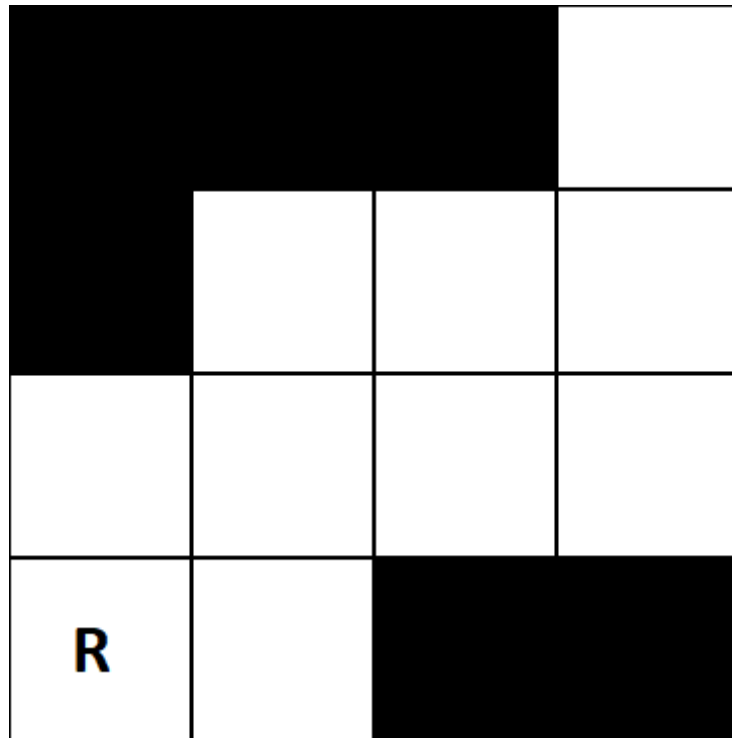
Όπως αναφέρθηκε στην προηγούμενη παράγραφο, κατά την αρχικοποίηση του αλγορίθμου, είναι απαραίτητη η εισαγωγή ορισμένων πληροφοριών. Οι πληροφορίες αυτές περιγράφουν την μορφολογία του περιβάλλοντος όπως διαστάσεις, εμπόδια, και ελεύθερος χώρος καθώς και πληροφορίες σχετικά με τα οχήματα όπως τύπος, διαθέσιμη ενέργεια και χωρητικότητα αποθήκευσης. Οι πληροφορίες αυτές εισάγονται στο σύστημα χρησιμοποιώντας ένα αρχείο εισόδου και παραμέτρους γραμμής εντολών.

3.3.1 Αρχείο εισόδου

Το αρχείο εισόδου περιγράφει την μορφολογία του περιβάλλοντος καθώς και τις αρχικές θέσεις των οχημάτων. Το αρχείο εισόδου έχει την μορφή απλού κειμένου (plain text). Τα σημεία στα οποία βρίσκονται αρχικά τα οχήματα σημειώνονται με 2, τα εμπόδια σημειώνονται με 1 ενώ τα ελεύθερα προσβάσιμα σημεία σημειώνονται

με 0. Για παράδειγμα, αν θέλουμε να περιγράψουμε ένα δισδιάστατο περιβάλλον 3x3 (σχήμα 3.4), με αρχική θέση ενός οχήματος στο σημείο (3,0) και εμπόδια στα σημεία (0,0), (1,0) (0,1), (0,2) (3,2) και (3,3), το αρχείο εισόδου θα έχει την μορφή που φαίνεται στην απεικόνιση (3.1).

Στην απεικόνιση 3.2 φαίνεται η συνάρτηση `ReadArrayFromFile` που διαβάζει την μορφολογία του περιβάλλοντος από ένα αρχείο κειμένου (text file). Η συνάρτηση επιστρέφει έναν τρισδιάστατο πίνακα ακέραιων αριθμών. Η πρώτη διάσταση υποδηλώνει τις γραμμές, η δεύτερη τις στήλες και η τρίτη το ύψος. Αξίζει να σημειωθεί πως στην περίπτωση που η τοπολογία του περιβάλλοντος είναι ένας δισδιάστατος χώρος, η τρίτη διάσταση θα έχει μέγεθος 1.



Σχήμα 3.4: Περιβάλλον δισδιάστατου χώρου 4x4. Στο σημείο (3,0) βρίσκεται αρχικά το όχημα. Τα εμπόδια βρίσκονται στα σημεία (0,0), (1,0) (0,1), (0,2) (3,2) και (3,3).

1,	1,	1,	0
1,	0,	0,	0
0,	0,	0,	0
2,	0,	1,	1

Απεικόνιση 3.1: Αρχείο εισόδου που περιγράφει το περιβάλλον του σχήματος 3.4

```
static int[] [] [] ReadArrayFromFile(int rows, int y, int z)
{
    int[] [] [] MyArray = new int[x] [] [];
    MyArray = InitializeArray(x, y, z);
    using (StreamReader R = new StreamReader(FileName))
    {
        int[] ThisLine;
        for (int height = 0; height < z; height++)
        {
            for (int b = 0; b < x; b++)
            {
                ThisLine = Array.ConvertAll(R.ReadLine()
                    .Split(','), s => int.Parse(s));
                for (int a = 0; a < y; a++)
                    MyArray[b][a][height] = ThisLine[a];
            }
        }
        return MyArray;
    }
}
```

Απεικόνιση 3.2: Συνάρτηση ReadArrayFromFile που διαβάζει την μορφολογία του περιβάλλοντος από ένα αρχείο κειμένου. Η συνάρτηση επιστρέφει έναν τρισδιάστατο πίνακα ο οποίος περιγράφει το περιβάλλον. Σε περίπτωση που το περιβάλλον είναι δισδιάστατο η τρίτη διάσταση του πίνακα είναι 1.

3.3.2 Παράμετροι γραμμής εντολών

Εκτός από το αρχείο εισόδου που περιγράφει τις διαστάσεις του περιβάλλοντος, τα εμπόδια και τις αρχικές θέσεις των οχημάτων, το πρόγραμμα υποστηρίζει και παραμέτρους γραμμής εντολών (command line arguments) για περαιτέρω παραμετροποίηση. Πιο συγκεκριμένα, οι παράμετροι γραμμής εντολών που υποστηρίζονται είναι:

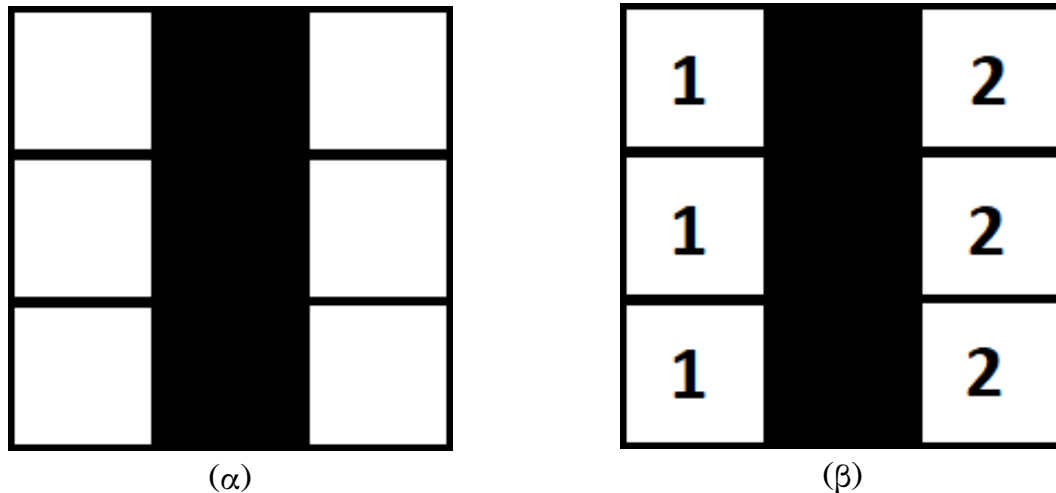
- Ο διακόπτης [-file file_name] που περιγράφει το όνομα του αρχείου εισόδου.
- Ο διακόπτης [-dim x,y,z] που περιγράφει τις διαστάσεις του περιβάλλοντος. Στην περίπτωση που πρόκειται για δισδιάστατο χώρο, το z είναι ίσο με 1.
- Ο διακόπτης [-ipos x,y,z] που περιγράφει τις αρχικές θέσεις των οχημάτων. Σε περίπτωση που έχουμε περισσότερα από ένα οχήματα, μπορούμε να διαχωρίσουμε τις αρχικές τους συντεταγμένες χρησιμοποιώντας τον χαρακτήρα ":". Για παράδειγμα, αν έχουμε ένα όχημα στην τοποθεσία (1,1,0) και ένα όχημα στην τοποθεσία (3,1,0) θα χρησιμοποιήσουμε τον διακόπτη ipos ως εξής: -ipos 1,1,0:3,1,0
- Ο διακόπτης [-vtype g/f] που περιγράφει τον τύπο του κάθε οχήματος. Πιο συγκεκριμένα, υποστηρίζονται οχήματα εδάφους (g) και οχήματα με δυνατότητα πλοήγησης σε όλα τα επίπεδα ύψους (f). Σε περίπτωση που έχουμε περισσότερα από ένα οχήματα, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των τιμών.
- Ο διακόπτης [-speed s] που περιγράφει την ταχύτητα του κάθε οχήματος. Η ταχύτητα του κάθε οχήματος είναι ένας θετικός ακέραιος αριθμός. Σε περίπτωση που έχουμε περισσότερα από ένα οχήματα, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των τιμών.
- Ο διακόπτης [-energy e] που περιγράφει την διαθέσιμη ενέργεια των οχημάτων (αν λαμβάνεται υπόψη). Η διαθέσιμη ενέργεια αναφέρεται στο πλήθος των χωροpixel που μπορεί ένα όχημα να προσπελάσει. Σε περίπτωση που ένα όχημα έχει απεριόριστη ενέργεια και δεν λαμβάνεται καθόλου υπόψη, χρησιμοποιείται ο ο διακόπτης με τιμή 0. Σε περίπτωση που έχουμε περισσότερα

από ένα οχήματα, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των τιμών.

- Ο διακόπτης [-energypos x,y,z] που περιγράφει τις συντεταγμένες στις οποίες βρίσκονται οι πηγές ενέργειας (αν υπάρχουν). Σε περίπτωση που έχουμε περισσότερες από μία πηγές ενέργειας, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των συντεταγμένων.
- Ο διακόπτης [-bstation x,y,z] που περιγράφει τα σημεία στα οποία βρίσκονται οι σταθμοί εκπομπής (αν υπάρχουν). Σε περίπτωση που έχουμε περισσότερους από έναν σταθμό εκπομπής, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των συντεταγμένων.
- Ο διακόπτης [-bsrange r] που περιγράφει την εμβέλεια του κάθε σταθμού εκπομπής. Σε περίπτωση που έχουμε περισσότερους από έναν σταθμό εκπομπής, χρησιμοποιείται ο χαρακτήρας ":" για το διαχωρισμό των τιμών.

3.4 Εύρεση συνδεδεμένων περιοχών

Σε ένα τρισδιάστατο περιβάλλον είναι πιθανό να υπάρχουν περιοχές που δεν είναι προσβάσιμες από το επίπεδο εδάφους, να είναι δηλαδή μεταξύ τους μη συνδεδεμένα συστατικά. Σε αυτήν την περίπτωση είναι απαραίτητο να βρούμε αυτά τα μη συνδεδεμένα συστατικά και ποια κελιά περιλαμβάνουν, καθώς και να βρούμε την συντομότερη διαδρομή που τα ενώνει χρησιμοποιώντας την τρίτη διάσταση, το ύψος. Για να λύσουμε το πρόβλημα, χρησιμοποιούμε τον αλγόριθμο που προτείνεται από τον Rosenfield κ.α. στο [39]. Πιο συγκεκριμένα, πρόκειται για έναν αλγόριθμο δύο περατώσεων [47] που εντοπίζει τα συνδεδεμένα συστατικά σε πίνακες δυαδικών δεδομένων. Ο αλγόριθμος κάνει δύο περάσματα πάνω από τον πίνακα. Το πρώτο εκχωρεί αρχικά προσωρινές ετικέτες σε κάθε κελί. Κατά το δεύτερο πέρασμα, πραγματοποιείται αντικατάσταση κάθε προσωρινής ετικέτας από την μικρότερη ετικέτα της κλάσης ισοδυναμίας της. Στον αλγόριθμο 1 βλέπουμε τον ψευδοκώδικα του αλγορίθμου. Στο σχήμα 3.5 βλέπουμε ένα παράδειγμα εντοπισμού των δύο μη συνδεδεμένων περιοχών ενός δισδιάστατου περιβάλλοντος.



Σχήμα 3.5: Στο σχήμα (α) βλέπουμε ένα δισδιάστατο περιβάλλον που χωρίζεται στην μέση από ένα τείχος εμποδίων. Στο (β) βλέπουμε πως ο αλγόριθμος εντοπίζει τις δύο μη συνδεδεμένες περιοχές.

Result: Connected Component Labeling

```

// First Pass
for (each element of the data by row) do
  for (each element of the data by column) do
    if (the cell is not in the background) then
      if (the cell has neighbors) then
        get the neighboring elements of the current element
        find the neighbor with the smallest label and assign it to the
        current element
        store the equivalence between neighboring labels
      if (the cell does not have neighbors) then
        uniquely label the current element and continue
      end
    end
  end
end
// Second Pass
for (each element of the data by row) do
  for (each element of the data by column) do
    if (the cell is not in the background) then
      relabel the element with the lowest equivalent label
    end
  end
end

```

Αλγόριθμος 1: Αλγόριθμος εύρεσης συνδεδεμένων στοιχείων στο επίπεδο εδάφους.

3.5 Εύρεση ελάχιστης απόστασης μεταξύ μη συνδεδεμένων περιοχών στο τρισδιάστατο επίπεδο

Αφού έχουν εντοπιστεί οι μη συνδεδεμένες περιοχές του περιβάλλοντος στο επίπεδο εδάφους, είναι απαραίτητη η εύρεση του μονοπατιού που τις ενώνει με το

μικρότερο δυνατό κόστος. Για την επίλυση αυτού του προβλήματος, χρησιμοποιήθηκε ο αλγόριθμος του Lee [40]. Πρόκειται για έναν αλγόριθμο βασισμένο στην αναζήτηση κατά πλάτος (Breadth First Search, BFS). Αναζήτηση κατά πλάτος είναι ένας αλγόριθμος για διάσχιση ή αναζήτηση σε δομές δεδομένων τύπου δέντρου ή γράφου. Η αναζήτηση ξεκινά από τη ρίζα του δέντρου ή από κάποιο αυθαίρετο κόμβο του γραφήματος, που μερικές φορές αναφέρεται ως "κλειδί αναζήτησης του" και διερευνά πρώτα τους γειτονικούς κόμβους, προτού μεταβεί στους γείτονες του επόμενου επίπεδο. Ακολουθεί αντίθετη στρατηγική σε σχέση με την αναζήτηση κατά βάθος, που διερευνά πρώτα τους κόμβους με το μεγαλύτερο βάθος. Στον αλγόριθμο 2 βλέπουμε τον ψευδοκώδικα του αλγορίθμου του Lee.

Result: Ψευδοκώδικας του αλγορίθμου του Lee

```
// Stage 1
// select start point and mark with 0
i = 0; // stage 2
do
    // mark all unlabeled neighbors of points marked with i with i+1
    i = i + 1
while target reached OR no points can be marked;
// Stage 3
// go to target point
do
    // go to next node that has a lower mark than the current node
    // add this node to path
    i = i + 1
while start point reached;
// Stage 4
// block the path for future wirings
// delete all marks
```

Αλγόριθμος 2: Αλγόριθμος εύρεσης μονοπατιού μεταξύ δύο σημείων στον τρισδιάστατο χώρο.

3.6 Διαμοιρασμός της συνολικής περιοχής σε υποπεριοχές για κάθε όχημα

Σε αυτό το στάδιο, χρησιμοποιώντας τον αλγόριθμο DARP (Divide Areas based on Robots initial Positions), η περιοχή ενδιαφέροντος χωρίζεται σε έναν αριθμό υποπεριοχών, που η κάθε μια αντιστοιχεί σε ένα συγκεκριμένο όχημα, έτσι ώστε να εξασφαλίζεται η πλήρης κάλυψη του συνολικής επιφάνειας και να εκμεταλλεύεται

πλήρως η δυναμική πολλαπλών οχημάτων.

Αρχικά, ο αλγόριθμος DARP χρησιμοποιεί μια αντιστοιχία ενός κελιού για κάθε όχημα. Διατηρείται ένας πίνακας αξιολόγησης E_i για κάθε όχημα. Αυτός ο πίνακας αξιολόγησης υποδεικνύει την απόσταση μεταξύ κάθε χωροpixel της περιοχής ενδιαφέροντος L και της αρχικής θέσης του κάθε οχήματος. Για να υπολογίσουμε αυτή την απόσταση χρησιμοποιούμε τον τύπο της ευκλείδειας απόστασης, εξ. 3.1.

$$D_v = \sqrt{(x_v - x_0)^2 + (y_v - y_0)^2}, \forall V \in L \quad (3.1)$$

όπου (x_v, y_v) υποδηλώνουν τις συντεταγμένες της αρχικής θέσης του οχήματος. Σε κάθε επανάληψη, υπολογίζεται ο πίνακας αντιστοίχισης A χρησιμοποιώντας την εξ. 3.2). Αυτός ο πίνακας αντιστοιχεί κάθε χωροpixel σε κάθε όχημα βάσει την απόσταση του κάθε χωροpixel από την αρχική θέση του κάθε οχήματος.

$$A_{x,y} = \min E_i, \forall (x, y) \in L \quad (3.2)$$

Κατά συνέπεια, η περιοχή του κάθε οχήματος L_i μπορεί να υπολογιστεί κατευθείαν χρησιμοποιώντας τον πίνακα αντιστοιχειών A χρησιμοποιώντας (εξ. 3.3).

$$L_i = \{(x, y) \in L : (x, y) = i\}, \forall \{i \in (1, \dots, n_r)\} \quad (3.3)$$

Επιπλέον, ο αριθμός των εκχωρημένων χωροpixel ανά όχημα k ορίζεται ως το πλήθος του συνόλου L_i set (εξ. 3.4).

$$k_i = |L_i|, \forall i \in \{1, \dots, n_r\} \quad (3.4)$$

Χρησιμοποιώντας τον πίνακα εκχωρήσεων A , ο DARP θα εκχωρήσει ένα χωροpixel σε μόνο ένα όχημα και κάθε χωροpixel θα αντιστοιχιστεί σε κάποια υποπεριοχή ενός οχήματος. Εννοείται, βέβαια, πως η αρχική θέση κάθε οχήματος εντασσεται

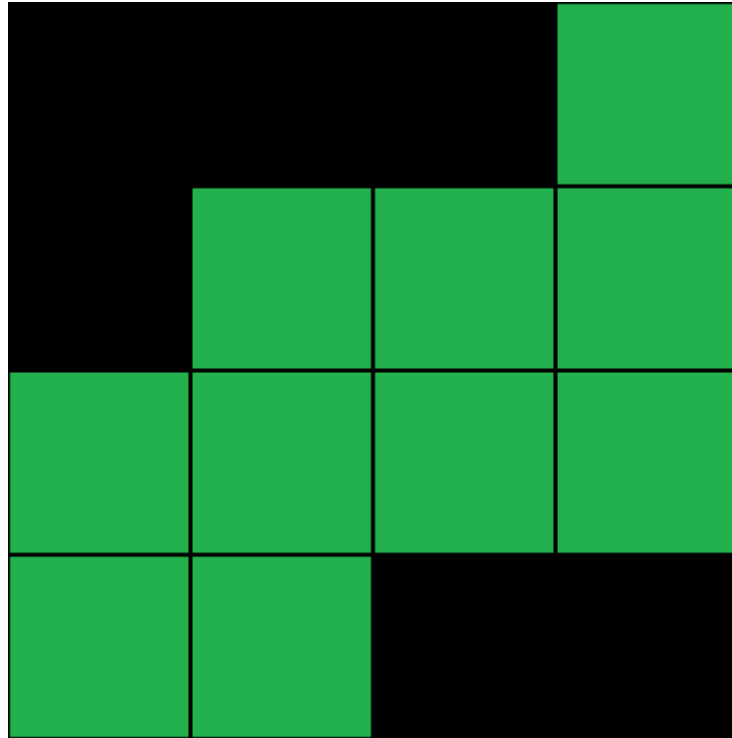
στην αντίστοιχη υποπεριοχή του.

Όπως έχουμε ήδη αναφέρει, οι πίνακες αξιολόγησης E_i αρχικά περιέχουν πληροφορίες απόστασης μεταξύ κάθε χωροpixel και οχήματος. Η βασική ιδέα του αλγορίθμου DARP είναι ότι κάθε πίνακας αξιολόγησης E_i μπορεί να τροποποιηθεί κατάλληλα, προκειμένου να εξισωθεί ο αριθμός των χωροpixel k_i για κάθε όχημα. Με άλλα λόγια, ο αλγόριθμος DARP επιχειρεί να δημιουργήσει υποπεριοχές ίσου μεγέθους για κάθε όχημα. Ωστόσο, δεδομένου ότι στην περίπτωσή μας κάθε όχημα μπορεί να έχει διαφορετική ταχύτητα κίνησης, απαιτούνται ορισμένες τροποποιήσεις στο συγκεκριμένο τμήμα. Αρχικά, διατηρείται ένας συντελεστής ταχύτητας WF_i για κάθε όχημα. Βάσει του συντελεστή ταχύτητας του κάθε οχήματος, προσδιορίζεται το ποσοστό της συνολικής περιοχής που θα διατεθεί σε κάθε όχημα. Τα οχήματα που μπορούν να κινούνται ταχύτερα θα αντιστοιχούν σε μια υψηλότερη τιμή WF_i . Για παράδειγμα, εάν το όχημα A (πράσινο) είναι δύο φορές πιο γρήγορα από το όχημα B (κόκκινο), στο όχημα A θα αντιστοιχεί $WF = 2$ ενώ στο όχημα B θα αντιστοιχεί $WF = 1$. Μετά την ολοκλήρωση της εκτέλεσης του τροποποιημένου αλγορίθμου DARP, αναμένεται ότι η υποπεριοχή που αντιστοιχεί στο όχημα A θα είναι διπλάσια της υποπεριοχής του οχήματος B. (σχήμα 3.7).

Ο αλγόριθμος DARP έχει ως έξοδο έναν πίνακα του οποίου οι τιμές υποδηλώνουν σε ποιο όχημα έχει ανατεθεί το κάθε χωροpixel. Στο σχήμα 3.6 βλέπουμε τον διαμορισμό της συνολικής περιοχής για το σχήμα 3.4.

3.7 Υπολογισμός Ελαχίστου Γεννητικού Δέντρου

Αφού έχει διαμοιραστεί ο συνολικός χώρος σε υποπεριοχές, μια για κάθε όχημα, είναι απαραίτητο να υπολογιστεί το Ελάχιστο Γεννητικό Δέντρο (ΕΓΔ) για κάθε υποπεριοχή, ώστε να υπολογιστεί στο επόμενο στάδιο το μονοπάτι κάλυψης της κάθε υποπεριοχής. Η διαδικασία αυτή φέρει μεγάλο υπολογιστικό κόστος. Είναι απαραίτητο λοιπόν να χρησιμοποιηθεί ένας αποδοτικός και γρήγορος αλγόριθμος. Οι τρεις κυρίαρχοι [18] αλγόριθμοι που επιλύουν το συγκεκριμένο πρόβλημα είναι: ο αλγόριθμος του Borůvka, [36] ο αλγόριθμος του Prim [12] και ο αλγόριθμος του Kruskal [26].



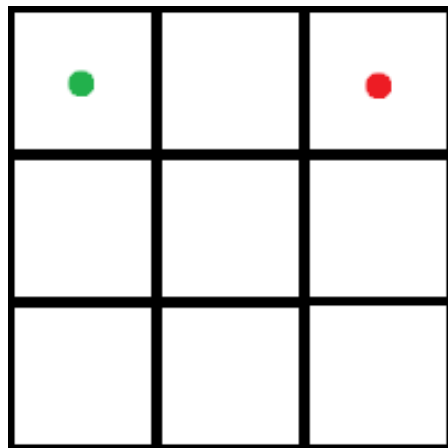
Σχήμα 3.6: Στο σχήμα βλέπουμε τον διαχωρισμό της συνολικής περιοχής χρησιμοποιώντας τον αλγόριθμο DARP για το σχήμα 3.4. Σε αυτήν την περίπτωση, έχουμε μόνο ένα όχημα. Συνεπώς, όλος ο χώρος θα ανατεθεί στο ένα και μοναδικό όχημα.

3.7.1 Αλγόριθμος του Borůvka

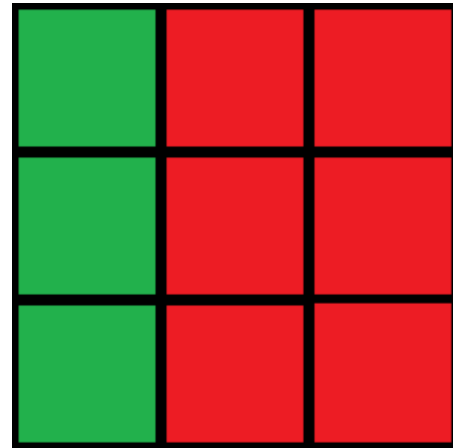
Ο αλγόριθμος του Borůvka δημιουργεί το ΕΓΔ από ένα δάσος δέντρων το οποίο αποτελείται αρχικά μόνο από τις κορυφές του γραφήματος G . Σε κάθε επανάληψη, ο αλγόριθμος εντοπίζει την ακμή με το ελάχιστο βάρος μεταξύ δυο διαφορετικών δέντρων του δάσους και συγχωνεύει τα υπολογισμένα δέντρα σε ένα μεγαλύτερο δέντρο. Ο αλγόριθμος συνεχίζει να εκτελείται μέχρι όλα τα δέντρα του δάσους έχουν ενωθεί δίνοντας το συνεκτικό τελικό ΕΓΔ [28]. Στον αλγόριθμο 3 δίνεται ο ψευδοκώδικας για τον αλγόριθμο του Borůvka. Η πολυπλοκότητα του αλγορίθμου σε σχέση με το πλήθος των ακμών και των κορυφών του γραφήματος είναι της τάξης $O(|E| * \log|V|)$.

3.7.2 Αλγόριθμος του Prim

Ο συγκεκριμένος αλγόριθμος, παρόλο που αρχικά προτάθηκε από τον Prim, V. Jarnik και E. Dijkstra, έμεινε γνωστός ως ο αλγόριθμος του Prim. Αρχικά, ξεκινάει από μια αυθαίρετη κορυφή του γράφου και δημιουργεί το δέντρο προσθέτοντας σε αυτό μια ακμή κάθε φορά. Ο αλγόριθμος τερματίζει σε $|V| - 1$ επαναλήψεις και σε



(α)



(β)

Σχήμα 3.7: Διαθέτουμε ένα γρήγορο όχημα, (κόκκινο) και ένα αργό όχημα (πράσινο) (α). Ο τροποποιημένος αλγόριθμος DARP θα διαμοιράσει κατάλληλα την συνολική αρχική περιοχή στα οχήματα βάσει της ταχύτητάς τους (β).

```

Function Boruvka (G: weighted undirected graph)
  T := forest consisting of G's vertexes and no edges
  While T is not connected
    For each  $T_i$  in T
      Find edge e with the lowest weight
      that connects  $T_i$  with  $T_i' \in T$ 
      Connect  $T_i$  and  $T_i'$  via e
  Return T

```

Αλγόριθμος 3: Αλγόριθμος του Boruvka.

κάθε μια προσθέτει στο δέντρο την πιο κοντινή εκτός δέντρου κορυφή, εκείνη δηλαδή με το ελάχιστο βάρος ακμής προς μια κορυφή του δέντρου που δεν έχει προστεθεί ήδη σε αυτό. Στον αλγόριθμο 4 δίνεται ο ψευδοκώδικας για τον αλγόριθμο του Prim.

```

Function Prim (G: weighted undirected graph)
  T := tree with vertex  $V_i \in V$ 
  For  $i = 1$  until  $i = |V| - 2$ 
    Find edge uv with the lowest weight so that
     $u \in V(T)$  and  $v \notin V(T)$ 
    T := T+uv
  Return T

```

Αλγόριθμος 4: Αλγόριθμος του Prim.

3.7.3 Αλγόριθμος του Kruskal

Αρχικά, ο αλγόριθμος του Kruskal ταξινομεί τις ακμές του γράφου κατά αύξον βάρος. Έπειτα, επιλέγει την ακμή με το ελάχιστο βάρος και την προσθέτει στο δέντρο. Σε κάθε επανάληψη, δημιουργεί ένα δάσος δέντρων ως υπό-γραφήματα του G , επιλέγει την ακμή με το μικρότερο βάρος και την προσθέτει στο αντίστοιχο δέντρο εφόσον η προσθήκη της δεν δημιουργεί κύκλο με αυτές που έχουν ήδη προστεθεί. Ο αλγόριθμος τερματίζει όταν έχουν προστεθεί $|V| - 1$ ακμές στο δέντρο. Στον αλγόριθμο 5 δίνεται ο ψευδοκώδικας για τον αλγόριθμο του Kruskal.

```
Function Kruskal (G: weighted undirected graph)
  T = empty subgraph of G
  Sort edges by weight in a list
  For  $i = 1$  until  $i = |V| - 1$ 
    Find edge  $e$  with the lowest weight
    If adding  $e$  does not create a circle
      then  $T = T + e$ 
    else remove  $e$  from list
  Return T
```

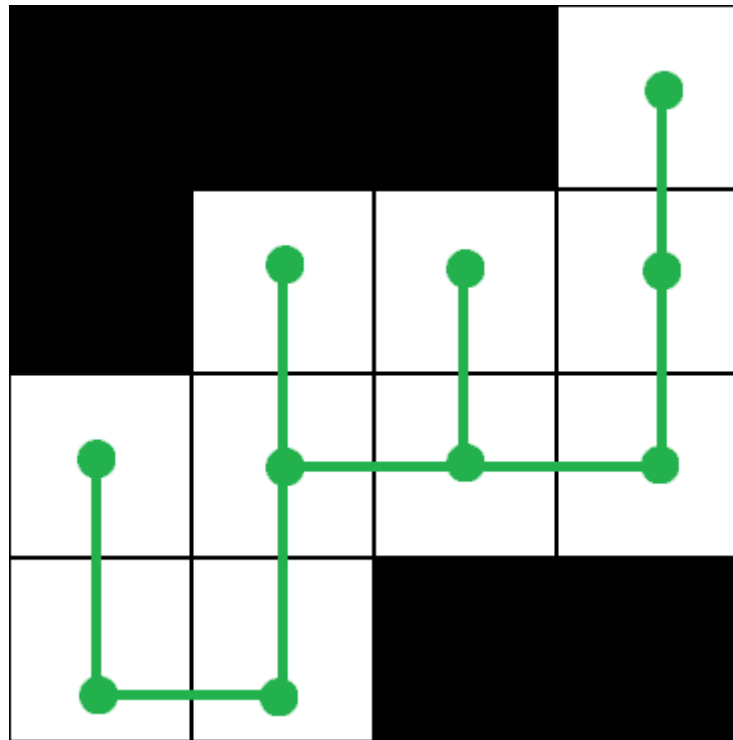
Αλγόριθμος 5: Αλγόριθμος του Kruskal.

Ο αλγόριθμος του Kruskal υπολογίζει το Ελάχιστο Γεννητικό Δέντρο με ανομοιόμορφο τρόπο, ο οποίος υποδεικνύεται από την κατάταξη των ακμών και των βαρών. Η πολυπλοκότητα του αλγορίθμου σε σχέση με το πλήθος των ακμών και κορυφών του μπορεί να είναι $O(|E| * \log|E|)$ για αραιά γραφήματα και $O(|E| * \log|V|)$ για πυκνά. Ενδείκνυται για περιπτώσεις στις οποίες τα βάρη των ακμών μπορούν να ταξινομηθούν σε μικρό χρόνο, όπως όταν είναι μικροί ακέραιοι αριθμοί ή όταν το γράφημα είναι αραιό, δεν περιέχει δηλαδή πολλές ακμές. Στο σχήμα 3.9 βλέπουμε σχηματικά την βηματική εκτέλεση του αλγορίθμου Kruskal.

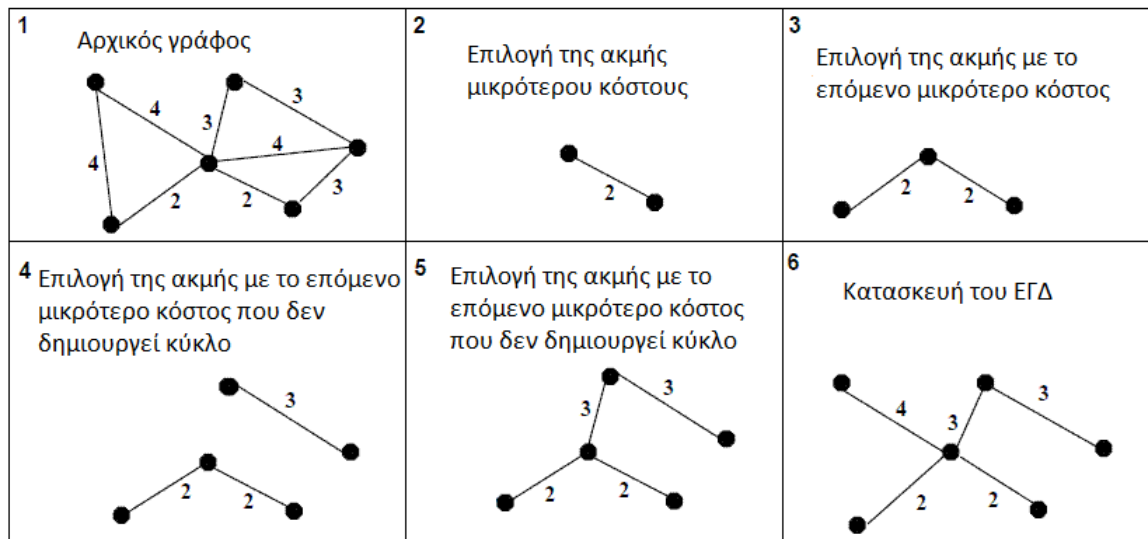
Στην παρούσα διπλωματική εργασία, για την κατασκευή των Ελαχίστων Γεννητικών Δέντρων χρησιμοποιήθηκε ο αλγόριθμος του Kruskal καθώς πειραματικά βρέθηκε ότι είναι 25% πιο αποδοτικός σε σχέση με τον αλγόριθμο του Prim και τον αλγόριθμο του Borůvka.

Ο αλγόριθμος του Kruskal ως έξοδο έχει το Ελάχιστο Γεννητικό Δέντρο της περιοχής του κάθε οχήματος σε μορφή γράφου. Για το σχήμα 3.6, το υπολογισμένο

Ελάχιστο Γεννητικό Δέντρο φαίνεται στο σχήμα 3.8.



Σχήμα 3.8: Στο σχήμα βλέπουμε το υπολογισμένο Ελάχιστο Γεννητικό Δέντρο για το σχήμα 3.6.



Σχήμα 3.9: Βηματική εκτέλεση του αλγορίθμου Kruskal.

3.7.4 Προσπέλαση ελαχίστου γεννητικού δέντρου

Αφού έχει υπολογιστεί το ελάχιστο γεννητικό δέντρο για την περιοχή που αντιστοιχεί στο κάθε όχημα, το επόμενο βήμα είναι να προσπελαστεί το κάθε ελάχιστο γεννητικό δέντρο ώστε να δημιουργηθεί το μονοπάτι κάλυψης του κάθε οχήματος.

Για αυτήν τη διαδικασία, χρησιμοποιήθηκε ο αλγόριθμος 6. Συνοπτικά, ο αλγόριθμος επιλέγει αρχικά ένα χωροpixel που συμπεριλαμβάνεται στο ΕΓΔ και ελέγχει τα γειτονικά χωροpixel. Έπειτα, επιλέγει το χωροpixel που βρίσκεται δεξιότερά του και προσθέτει τις συντεταγμένες του σε μια λίστα. Αυτό συνεχίζεται μέχρις ότου να προσπελαστούν όλα τα χωροpixel. Τέλος, επιστρέφεται η λίστα με τις συντεταγμένες των σημείων που προσπελάστηκαν. Στο σχήμα 3.10 βλέπουμε τον υπολογισμό του μονοπατιού κάλυψης χρησιμοποιώντας τον αλγόριθμο.

Function STC

L = an empty list of vortiles

T = a random vortile that belongs to the MST

do

 find T 's closest neighbor n in
 the counter clockwise direction

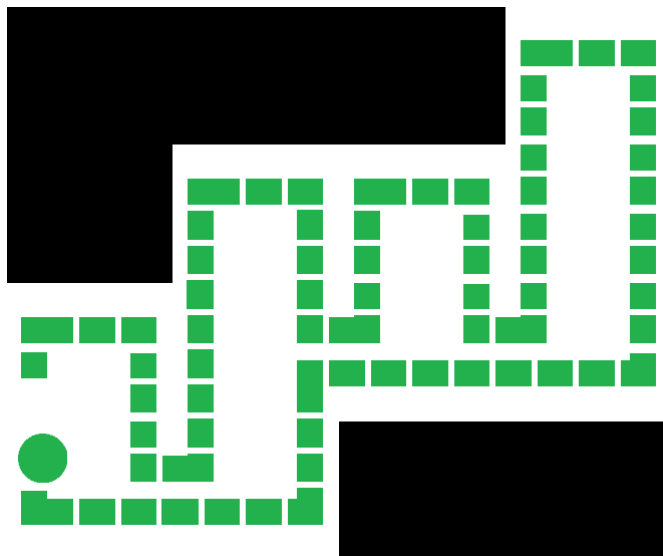
 add n in L

$T = n$

while $T \neq \emptyset$

Return L

Αλγόριθμος 6: Αλγόριθμος προσπέλασης ΕΓΔ για τη δημιουργία αρχικού μονοπατιού κάλυψης.



Σχήμα 3.10: Βλέπουμε το υπολογισμένο μονοπάτι κάλυψης χώρου για το σχήμα 3.8.

3.8 Υπολογισμός τελικού μονοπατιού

Αφού δημιουργηθεί το μονοπάτι κάλυψης χώρου χρησιμοποιώντας τον αλγόριθμο 6, πραγματοποιείται η προσπέλαση του. Κατά την προσπέλαση, διατηρείται ένας μετρητής C που μετράει το πλήθος των χωροpixel που έχουν επισκευθεί μέχρι στιγμής. Επίσης, διατηρείται ένας αθροιστής S που αθροίζει το μέγεθος των δεδομένων που έχουν συλλεχθεί μέχρι στιγμής. Σε κάθε βήμα, χρησιμοποιώντας τον αλγόριθμο 2, υπολογίζεται η συντομότερη διαδρομή CED (*ClosestEnergyDistance*) και το κόστος αυτής (C_{CED}) που συνδέει το συγκεκριμένο χωροpixel από την το σύνολο των χωροpixel στα οποία υπάρχει διαθέσιμη πηγή ενέργειας. Επιπλέον, υπολογίζεται η συντομότερη διαδρομή CDD (*ClosestDataDistance*) και το κόστος αυτής (C_{CDD}) που συνδέει το συγκεκριμένο χωροpixel από το σύνολο όλων των χωροpixel στα οποία εκπέμπουν οι σταθμοί εκπομπής. Έτσι, όταν η εξ. 3.5 ή η εξ. 3.6 πλησιάζουν στο 0, σταματάει προσωρινά η προσπέλαση του Ελαχίστου Γεννητικού Δέντρου και το όχημα κατευθύνεται προς την πηγή ενέργειας ή προς το συντομότερο χωροpixel στο οποίο υπάρχει εκπομπή ενός σταθμού. Για την κατεύθυνση του οχήματος χρησιμοποιούνται οι διαδρομές CED και CDD που υπολογίστηκαν προηγουμένως. Σε περίπτωση που ταυτοχρονα μηδενίζουν και οι δύο εξισώσεις, δίνεται προτεραιότητα στην πηγή ενέργειας. Έτσι, ο αλγόριθμος εγγυάται: (α) πως το κάθε όχημα θα επιλέξει να χρησιμοποιήσει την πηγή ενέργειας την κατάλληλη στιγμή, (β) πως κανένα όχημα δεν σταματήσει στην μέση της κάλυψης χώρου λόγω έλλειψης ενέργειας και (γ) πως τα οχήματα θα επισκέπτονται χωροpixel εντός εμβέλειας κάποιου σταθμού εκπομπής όταν αυτό είναι απαραίτητο, δηλαδή όταν ο διαθέσιμος αποθηκευτικός χώρος πλησιάζει στο 0.

$$E = TotalEnergy - C - C_{CED} \quad (3.5)$$

$$D = TotalCapacity - C \cdot DataCollectedPerVotile - C_{CDD} \cdot DataCollectedPerVotile \quad (3.6)$$

3.9 Μετρικές κώδικα

Για την υλοποίηση του εργαλείου που παρουσιάστηκε στην παρούσα διπλωματική εργασία, δημιουργήθηκαν συνολικά 23 τα οποία βρίσκονται μέσα σε 5 φακέλους. Τα αρχεία αυτά αποτελούνται από: πηγαία αρχεία κώδικα C#, βιβλιοθήκες .NET DLL (Dynamic-Link Library) καθώς και 2 εικονίδια. Στα μετρικά που παρουσιάζονται στον πίνακα 3.1 δεν προσμετρώνται τα εκτελέσιμα αρχεία (binary files) που δημιουργούνται αυτόματα κατά την μεταγλώττιση.

τύπος	πλήθος	μέγεθος	σχόλια	γραμμές κώδικα
πηγαία αρχεία κώδικα C#	13	96000	174	2142
βιβλιοθήκες .NET DLL	8	245000	425	5479
εικονίδια	2	122000	-	-
Σύνολο	23	463000	599	7621

Πίνακας 3.1: Μετρικές αρχείων κώδικα.

3.10 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν τα μέρη που συγκροτούν το εργαλείο και η λειτουργία του καθενός από αυτά. Η παρουσίαση συμπληρώθηκε με την αναπαράσταση των αλγορίθμων, σε ψευδοκώδικα, όπου αυτό κρίθηκε απαραίτητο. Επιπλέον, δόθηκαν ορισμένες μετρικές κώδικα, όπως μέγεθος αρχείων και αριθμός γραμμών κώδικα. Στο επόμενο κεφάλαιο παρουσιάζονται παραδείγματα της εκτέλεσης του εργαλείου χρησιμοποιώντας συνθετικά περιβάλλοντα.

Κεφάλαιο 4

Πειραματικές μετρήσεις

Στο κεφάλαιο αυτό παρουσιάζεται η βηματική εκτέλεση του εργαλείου σε δύο συνθετικά περιβάλλοντα. Παρουσιάζονται τα αρχεία εισόδου, οι παράμετροι γραμμής εντολών που χρησιμοποιήθηκαν καθώς και οι εισοδοί και οι έξοδοι των διαφόρων σταδίων του εργαλείου.

4.1 Παράδειγμα εκτέλεσης σε δισδιάστατο συνθετικό περιβάλλον

Στο παράδειγμα αυτό παρουσιάζεται ένα συνθετικό περιβάλλον δισδιάστατου χώρου διαστάσεων (4x4). Τα εμπόδια βρίσκονται στα σημεία (0,0), (0,1), (0,2), (3,1), (3,2) και (3,3). Το ένα και μοναδικό όχημα βρίσκεται στη θέση (1,0). Το αρχείο εισόδου που διαβάζει ο αλγόριθμος έχει την ονομασία “example1.txt“. Στην απεικόνιση 4.1 βλέπουμε την μορφή του αρχείου εισόδου.

1,	1,	1,	0
2,	0,	0,	0
0,	0,	0,	0
0,	1,	1,	1

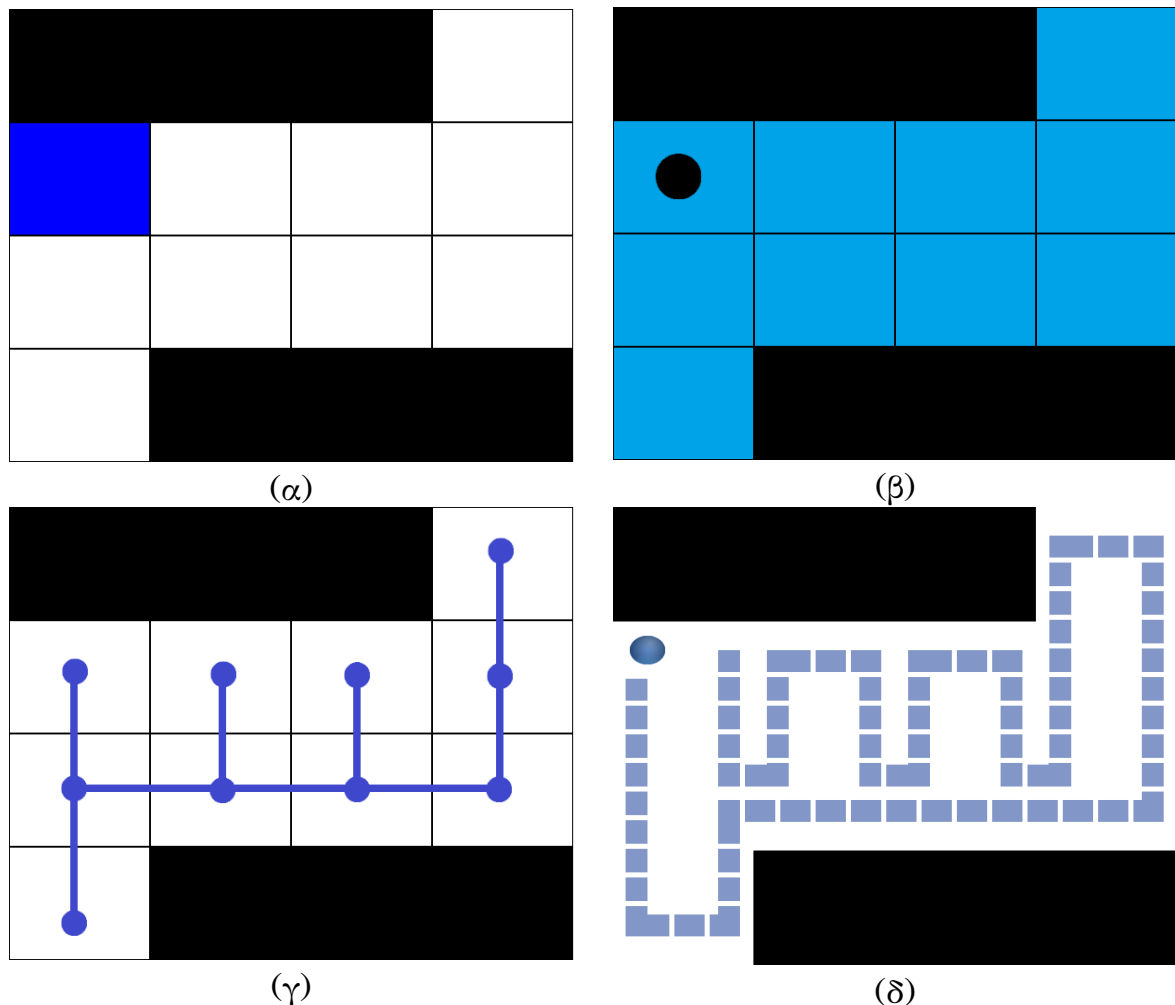
Απεικόνιση 4.1: Αρχείο εισόδου για το παράδειγμα εκτέλεσης δισδιάστατου χώρου

Έπειτα, εκτελούμε το εργαλείο εύρεσης μονοπατιού κάλυψης χρησιμοποιώντας την εντολή που φαίνεται στην απεικόνιση 4.2.

```
CPP3D -file example1.txt -dim 4,4,1 -ipos 1,0,0 -e 0
```

Απεικόνιση 4.2: Εντολή εκτέλεσης του εργαλείου για το παράδειγμα 4.1

Η έξοδος του αλγορίθμου, δηλαδή το μονοπάτι κάλυψης του χώρου αποθηκεύεται σε ένα νέο αρχείο με όνομα “example1_results.txt”. Η γραφική αναπαράσταση των επιμέρους τμημάτων του εργαλείου φαίνεται στο σχήμα 4.1.



Σχήμα 4.1: Στο (α) βλέπουμε την γραφική απεικόνιση του περιβάλλοντος. Στο (β) βλέπουμε την έξοδο του αλγορίθμου διαμοιρασμού της συνολικής περιοχής στα οχήματα. Στην συγκεκριμένη περίπτωση, όλη η περιοχή ανήκει στο ένα και μοναδικό όχημα. Στο (γ) βλέπουμε την κατασκευή του Ελαχίστου Γεννητικού Δέντρου. Τέλος, στο (δ) βλέπουμε την κατασκευή του μονοπατιού κάλυψης χώρου.

4.2 Παράδειγμα εκτέλεσης σε τρισδιάστατο συνθετικό περιβάλλον

Στο παράδειγμα αυτό παρουσιάζεται ένα συνθετικό περιβάλλον τρισδιάστατου χώρου διαστάσεων (6x6). Τα εμπόδια βρίσκονται στα σημεία (0,0,0), (0,1,0), (0,2,0), (0,3,0), (1,0,0), (1,1,0), (1,2,0), (1,3,0), (2,2,0), (2,3,0), (2,4,0), (2,5,0), (3,5,0), (5,1,0), (5,2,0) και (5,5,0). Το πρώτο όχημα βρίσκεται στην θέση (4,0,0) και το δεύτερο όχημα βρίσκεται στη θέση (1,5,0). Το αρχείο εισόδου που διαβάζει ο αλγόριθμος

έχει την ονομασία “example2.txt“. Στην απεικόνιση 4.3 βλέπουμε την μορφή του αρχείου εισόδου.

1,	1,	1,	1,	0,	0
1,	1,	1,	1	0,	2
0,	0,	1,	1,	1,	1
0,	0,	0,	0,	0,	0
0,	1,	1,	0,	0,	1
0,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0

Απεικόνιση 4.3: Αρχείο εισόδου για το παράδειγμα εκτέλεσης τρισδιάστατου χώρου

Έπειτα, εκτελούμε το εργαλείο εύρεσης μονοπατιού κάλυψης χρησιμοποιώντας την εντολή που φαίνεται στην απεικόνιση 4.4.

```
CPP3D -file example2.txt -dim 6,6,2 -ipos 4,0,0:1,5,0 -e 0:0  
-vtype g:f
```

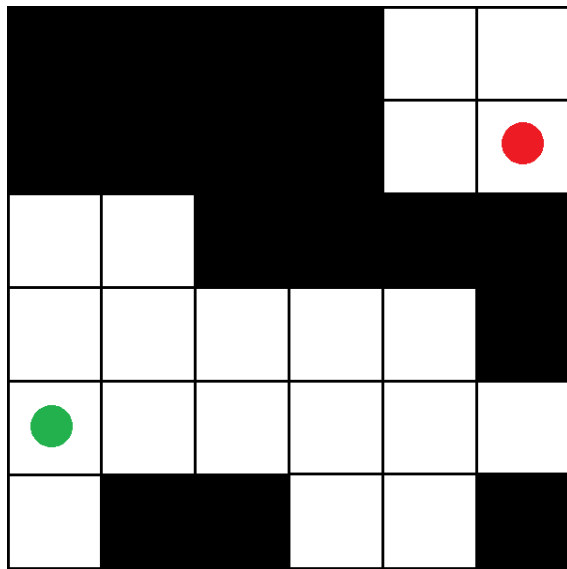
Απεικόνιση 4.4: Εντολή εκτέλεσης του εργαλείου για το παράδειγμα 4.2

Η έξοδος του αλγορίθμου, δηλαδή το μονοπάτι κάλυψης του χώρου αποθηκεύεται σε ένα νέο αρχείο με όνομα “example1_results.txt“. Η γραφική αναπαράσταση των επιμέρους τμημάτων του εργαλείου φαίνεται στο σχήμα 4.1.

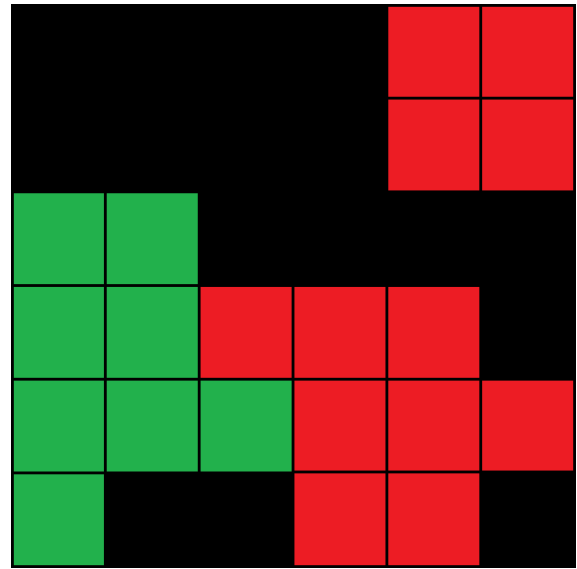
4.3 Χρόνος εκτέλεσης εργαλείου

Για την εύρεση του βέλτιστου μονοπατιού κάλυψης στο παράδειγμα δισδιάστατου χώρου χρησιμοποιώντας ένα όχημα χρειάστηκαν μόλις 3.7 δευτερόλεπτα. Στο παράδειγμα τρισδιάστατου χώρου χρησιμοποιώντας δύο οχήματα χρειάστηκαν 8.6 δευτερόλεπτα. Το υλικό (hardware) που χρησιμοποιήθηκε αποτελείται από:

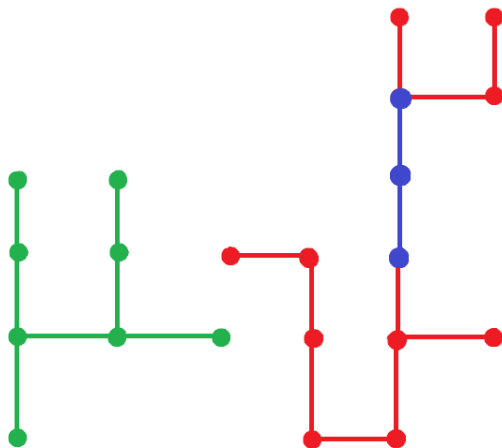
- CPU Intel Core 2 Quad Q9550 2.83 GHz



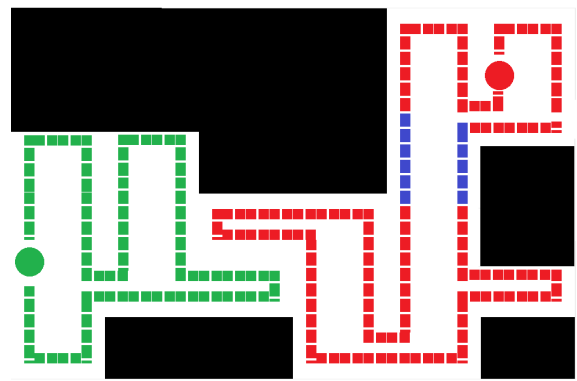
(α)



(β)



(γ)



(δ)

Σχήμα 4.2: Στο (α) βλέπουμε την γραφική απεικόνιση του περιβάλλοντος. Στο (β) βλέπουμε την έξοδο του αλγορίθμου διαμοιρασμού της συνολικής περιοχής στα οχήματα. Στο (γ) βλέπουμε την κατασκευή των Ελαχίστων Γεννητικών Δέντρων. Τέλος, στο (δ) βλέπουμε την κατασκευή των μονοπατιών κάλυψης χώρου. Τα σημεία του μονοπατιού του κόκκινου οχήματος που εμφανίζονται με μπλε χρώμα υποδεικνύουν πως το όχημα θα προσπεράσει το εμπόδιο που βρίσκεται σε εκείνο το σημείο πετώντας από πάνω.

-
- RAM DDR2 800 MHz
 - Motherboard Foxcon G31MX
 - GPU Nvidia GeForce GTX 750 Ti 2GB

4.4 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν ορισμένα πειραματικά αποτελέσματα που αφορούν τα μονοπάτια κάλυψης χώρου που παράγονται από το εργαλείο μας. Επιλέχθηκαν δύο χαρακτηριστικά περιβάλλοντα δισδιάστατου και τρισδιάστατου χώρου με ένα και δύο οχήματα αντίστοιχα. Επίσης, δόθηκαν και οι χρόνοι εκτέλεσης του εργαλείου για τα δύο παραδείγματα. Στο επόμενο κεφάλαιο, παρουσιάζονται τα συμπεράσματα, η εκδοχή ίδρυσης επιχείρησης καθώς και πιθανές μελλοντικές επεκτάσεις του εργαλείου.

Κεφάλαιο 5

Συμπεράσματα και μελλοντικές βελτιώσεις

Συνοψίζοντας, καταγράφουμε τα συμπεράσματα από την εκπόνηση της παρούσας διπλωματικής εργασίας. Επιπλέον, αναφέρουμε την εκδοχή ίδρυσης επιχείρησης καθώς και πιθανές μελλοντικές επεκτάσεις που μπορούν να υλοποιηθούν στο εργαλείο που παρουσιάστηκε.

5.1 Συμπεράσματα

Το εργαλείο που αναπτύχθηκε στην παρούσα διπλωματική εργασία έχει σαν στόχο την εύρεση του μονοπατιού κάλυψης για αυτόνομα οχήματα σε ορισμένο και γνωστό χώρο. Ο χώρος μπορεί να είναι δισδιάστατος ή τρισδιάστατος, ενώ υποστηρίζονται οχήματα εδάφους και οχήματα με δυνατότητα πλοήγησης στο κατακόρυφο άξονα. Επιπλέον, για την κάλυψη του χώρου μπορούν να χρησιμοποιηθούν ένα ή περισσότερα οχήματα. Στην περίπτωση που υπάρχουν περισσότερα από ένα οχήματα, το πρόγραμμα διαμοιράζει τον συνολικό χώρο στα διαθέσιμα οχήματα ανάλογα με τις δυνατότητές τους. Κατά την δημιουργία του μονοπατιού κάλυψης λαμβάνονται υπόψη διάφοροι περιορισμοί όπως η διαθέσιμη ενέργεια του κάθε οχήματος, ο αποθηκευτικός του χώρος και η ταχύτητά του. Τα οφέλη που προκύπτουν από την υλοποίηση αυτού του εργαλείου είναι πολλά και βρίσκουν εφαρμογή σε διάφορους τομείς, όπως η γεωργία, η χαρτογράφηση, η εξερεύνηση του βυθού της θάλασσας και η εξερεύνηση του διαστήματος.

Αρχικά, ο τομέας της γεωργικής δραστηριότητας μπορεί να έχει πολλαπλά οφέλη από την χρήση των αυτό-δρομολογούμενων αυτόνομων οχημάτων. Η προώθηση και

η χρήση αυτών μπορεί να γίνει σύμμαχος του παραγωγού, βοηθώντας τον να διαχειριστεί και να ελέγξει με μεγαλύτερη ακρίβεια τον χώρο του και επιτρέποντας του να επέμβει με αμεσότητα και ακρίβεια όπου απαιτείται. Επίσης, του παρέχει οικονομικά οφέλη, καθώς συμβάλλει στη μείωση του κόστους και του χρόνου εργασίας. Ταυτόχρονα, με τη μείωση των γεωργικών υπολειμμάτων από την ορθολογικότερη χρήση των λιπασμάτων και των φυτοφαρμάκων, μειώνεται η περιβαλλοντική ρύπανση, δίνοντας έτσι λύση σε ένα πρόβλημα που απασχολεί ολόκληρο τον πλανήτη.

Τα μη επανδρωμένα ιπτάμενα οχήματα (drones) βρίσκουν, επίσης, εφαρμογή σε πολλούς τομείς. Παρέχουν υψηλής ανάλυσης γεωγραφικά δεδομένα και πρόσβαση σε δύσβατες περιοχές, με εικόνες υψηλής ανάλυσης και ακρίβειας. Και αυτά, λοιπόν, αποτελούν χρήσιμα εργαλεία σε ένα μεγάλο φάσμα δραστηριοτήτων, στρατιωτικές επιχειρήσεις, τηλεοπτικές και κινηματογραφικές παραγωγές, τοπογραφικές αποτυπώσεις, ασφάλεια χώρων.

Η χρήση αυτόνομων οχημάτων και των ρομποτικών μηχανισμών μπορεί, ακόμη, να βρει εφαρμογή και να βελτιώσει σε μεγάλο βαθμό στην εξερεύνηση και χαρτογράφηση των υποθαλάσσιων περιοχών. Ένα μη επανδρωμένο υποθαλάσσιο όχημα, εξοπλισμένο με κάμερες υψηλής ανάλυσης και αισθητήρες μεγάλης ακρίβειας, μπορεί να πλοηγηθεί με τον βέλτιστο τρόπο, καταναλώνοντας την ελάχιστη δυνατή ενέργεια, και να μας παρέχει εικόνες και μετρήσεις για τις υποβρύχιες συνθήκες.

Τα αυτο-δρομολογούμενα αυτόνομα οχήματα αποτελούν ένα ιδιαίτερα χρήσιμο εργαλείο στην εξερεύνηση και την κατανόηση περιοχών του διαστήματος ή άλλων πλανητών. Το εργαλείο που αναπτύχθηκε στην παρούσα διπλωματική εργασία μπορεί να χρησιμοποιηθεί σε αυτόνομα οχήματα διαστημικών αποστολών για την λήψη μετρήσεων, φωτογραφιών και δειγμάτων. Έτσι, επιτυγχάνεται μεγάλη συμβολή στην εξέλιξη της επιστημονικής έρευνας και στην διεξαγωγή επιστημονικών μελετών.

5.2 Προοπτική δημιουργίας επιχείρησης

Σε αυτήν την παράγραφο εξετάζεται η περίπτωση δημιουργίας μίας πιθανής επιχείρησης βασιζόμενη στην εύρεση μονοπατιού κάλυψης χρησιμοποιώντας το εργαλείο που αναπτύχθηκε στην παρούσα διπλωματική εργασία. Εξετάζονται οι πιθανές μορφές με τις οποίες μπορεί να προσφερθεί η υπηρεσία και γίνεται ανάλυση SWOT.

5.2.1 Μοντέλο επιχείρησης

Το SaaS (Software as a Service) είναι ένα μοντέλο επιχείρησης στο οποίο ο πελάτης δεν αγοράζει λογισμικό από μία επιχείρηση, αλλά την παροχή υπηρεσιών του λογισμικού. Συνήθως, ο πελάτης πληρώνει μηνιαία/ετησίως με την μορφή συνδρομής. Ένα σημαντικό πλεονέκτημα αυτού του μοντέλου είναι το γεγονός ότι ο πελάτης δεν χρειάζεται να εγκαταστήσει και να πραγματοποιεί συντήρηση στο πρόγραμμα. Η υπηρεσία βρίσκεται συνεχώς διαθέσιμη διαδικτυακά και ο πελάτης μπορεί να έχει πρόσβαση σε αυτή ανα πάσα στιγμή. Έτσι, η υπηρεσία βρίσκεται συνεχώς στην τελευταία έκδοση και ο πελάτης απολαμβάνει μια καλύτερη εμπειρία χρήσης. Σε αυτήν την περίπτωση, είναι σημαντική η ύπαρξη ενός web API (Application Programming Interface), δηλαδή ενός συνόλου λειτουργιών που μπορούν να εκτελούνται χωρίς την ανάγκη αλληλεπίδρασης με φυσικό πρόσωπο αλλά από άλλο πρόγραμμα. Δεδομένου ότι οι αλγόριθμοι που παρουσιάζονται στην παρούσα διπλωματική εργασία έχουν μεγάλο υπολογιστικό κόστος, κρίνεται απαραίτητη η ύπαρξη διακομιστών υψηλών επιδόσεων. Επιπλέον, είναι απαραίτητη η ύπαρξη συνοδευτικών εγγράφων που περιγράφουν αναλυτικά τις λειτουργίες της υπηρεσίας καθώς και τους τρόπους πρόσβασης στο API.

Ένα άλλο πιθανό μοντέλο παράδοσης λογισμικού στο οποίο θα μπορούσε να βασιστεί μια επιχείρηση είναι το αδειοδοτημένο μοντέλο (licensed model). Στο μοντέλο αυτό, η επιχείρηση πουλάει το λογισμικό σε εκτελέσιμη δυαδική μορφή. Σε αυτήν την περίπτωση, είναι απαραίτητη η ύπαρξη κατάλληλων εγγράφων και οδηγιών που περιγράφουν την λειτουργία αλλά και τον τρόπο επέκτασης του προγράμματος.

5.2.2 Ανάλυση SWOT

Η ανάλυση SWOT είναι ένα εργαλείο στρατηγικού σχεδιασμού το οποίο χρησιμοποιείται για την ανάλυση του εσωτερικού και εξωτερικού περιβάλλοντος μίας επιχείρησης, όταν η επιχείρηση πρέπει να λάβει μία απόφαση σε σχέση με τους στόχους που έχει θέσει ή με σκοπό την επίτευξή τους. Το αρχικόλεξο SWOT προκύπτει από τις αγγλικές λέξεις: Strengths, Weaknesses, Opportunities, Threats (αντίστοιχα στα ελληνικά: Δυνατά σημεία, Αδύνατα σημεία, Ευκαιρίες, Απειλές). Για την περίπτωση σύστασης μιας επιχείρησης με βάση το το εργαλείο που παρουσιάστηκε στην παρούσα διπλωματική εργασία, δίνεται η ανάλυση SWOT στον πίνακα 5.1.

<p style="text-align: center;">Δυνατά σημεία</p> <p>Cloud εφαρμογή Ευκολία στη χρήση Εύκολη μελλοντική επέκταση</p>	<p style="text-align: center;">Αδύνατα σημεία</p> <p>Έλλειψη API Απαραίτητη ύπαρξη διακομιστών υψηλών επιδόσεων</p>
<p style="text-align: center;">Ευκαιρίες</p> <p>Αυτοματοποίηση εύρεσης μονοπατιού κάλυψης Τεχνολογική τάση εφαρμογών νέφους (cloud computing)</p>	<p style="text-align: center;">Απειλές</p> <p>Μεγάλες εταιρίες στον χώρο</p>

Πίνακας 5.1: SWOT ανάλυση για σύσταση επιχείρησης

5.3 Μελλοντικές επεκτάσεις

Με την ανάπτυξη του εργαλείου που παρουσιάστηκε στην παρούσα διπλωματική εργασία, τέθηκαν οι βάσεις για μία σειρά βελτιώσεων και επεκτάσεων που μπορούν να υλοποιηθούν ώστε να αυξηθεί η αποδοτικότητα, να συμπληρωθεί η χρηστικότητα και να γίνουν οι συνθήκες εκτέλεσης του εργαλείου ακόμη πιο ρεαλιστικές. Οι βελτιώσεις αυτές είναι χρήσιμες όχι μόνο για τον τελικό χρήστη, αλλά και για τους ερευνητές που θα επιλέξουν να συνεχίσουν την ανάπτυξη του παρόντος εργαλείου. Ορισμένες προτάσεις έχουν επίσης ως στόχο τον εκσυγχρονισμό του κώδικα ώστε να προχωράει παράλληλα με τις τεχνολογικές τάσεις και να είναι σε θέση να ανταγωνίζεται παρόμοια εργαλεία που χρησιμοποιούν άλλες τεχνολογίες και αλγορίθμους.

5.3.1 Προσθήκη επιπλέον περιορισμών

Μια κατηγορία μελλοντικών επεκτάσεων αποτελεί η προσθήκη νέων περιορισμών στο πρόγραμμα. Αυτήν τη στιγμή, οι περιορισμοί των οχημάτων που υποστηρίζονται περιλαμβάνουν την ταχύτητα, τον αποθηκευτικό χώρο, την διαθέσιμη ενέργεια και τον τύπο του οχήματος. Ως ταχύτητα του οχήματος, ορίζεται η ικανότητα του να μετακινείται στο χώρο, δηλαδή να μετακινείται από ένα χωροpixel σε ένα άλλο. Ως αποθηκευτικός χώρος ορίζεται η ικανότητα του οχήματος να αποθηκεύει δεδομένα που έχει συλλέξει σχετικά με το περιβάλλον μέχρις ότου να επισκευθεί ένα χωροpixel εντός εμβέλειας κάποιου σταθμού εκπομπής. Ως διαθέσιμη ενέργεια ορίζονται τα χωροpixel στα οποία μπορεί να πλοηγηθεί το όχημα πρώτου χρειαστεί

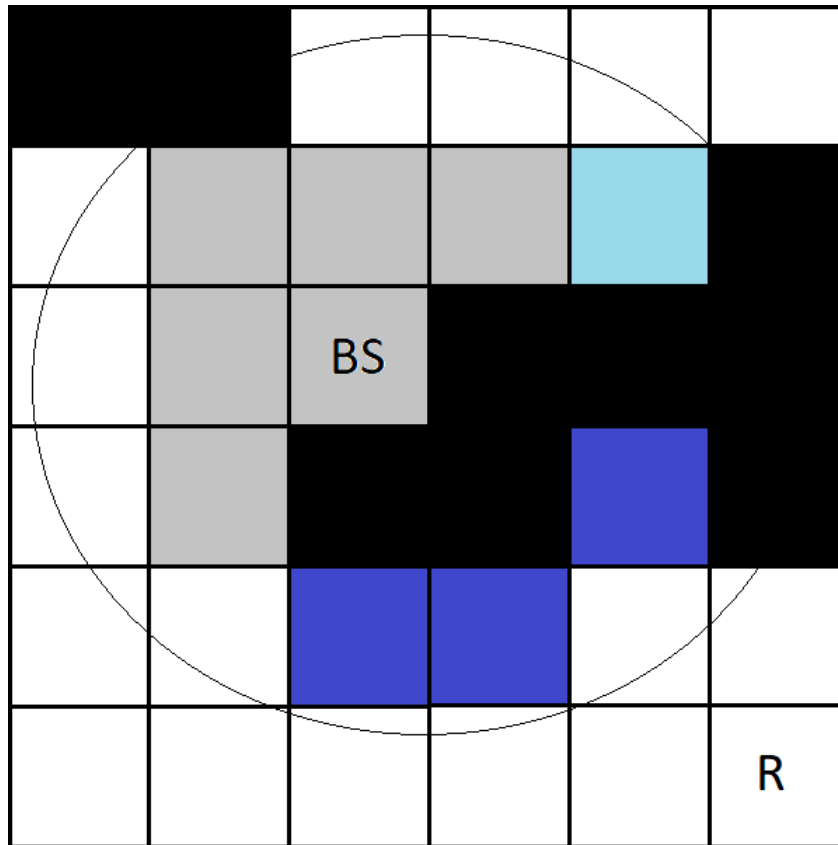
να επισκεφθεί μια πηγή ενέργειας. Τέλος, οι δύο υποστηριζόμενοι τύποι οχημάτων είναι τα οχήματα εδάφους και τα οχήματα που μπορούν να πλοηγούνται σε όλα τα επίπεδα ύψους.

Όσον αφορά την διαθέσιμη ενέργεια των οχημάτων, μία πιθανή επέκταση αποτελεί η περαιτέρω παραμετροποίηση της. Πιο συγκεκριμένα, θα μπορεί να ορίζεται η κατανάλωση της κάθε ξεχωριστής κίνησης των οχημάτων. Για παράδειγμα, θα μπορεί να οριστεί μεγαλύτερη κατανάλωση ενέργειας στην δεξιά από ότι στην αριστερή στροφή ενός οχήματος. Έτσι, το εργαλείο θα καλείται να βρει το μονοπάτι κάλυψης του συνολικού χώρου ελαχιστοποιώντας τις δεξιές στροφές και μεγιστοποιώντας τις αριστερές στροφές. Αυτή η επέκταση θα ήταν εξαιρετικά σημαντική για ορισμένους ρομποτικούς μηχανισμούς που πραγματοποιούν συγκεκριμένες κινήσεις πιο εύκολα σε σχέση με άλλες.

Στην ίδια κατηγορία βελτιώσεων ανήκει και η τροποποίηση του εργαλείου ώστε να λαμβάνεται υπόψη η επιτάχυνση των οχημάτων. Αυτήν τη στιγμή, η ταχύτητα του κάθε οχήματος θεωρείται σταθερή και αμετάβλητη, κάτι που απέχει από τις συνθήκες που επικρατούν στον πραγματικό κόσμο. Το εργαλείο θα μπορούσε να τροποποιηθεί κατάλληλα ώστε η ταχύτητα του κάθε οχήματος να μεταβάλεται ανάλογα με την επιτάχυνση του. Έτσι, όταν ένα όχημα κινείται σε ευθεία γραμμή θα μπορεί να καλύπτει περισσότερο χώρο συγκριτικά με την κίνηση κοντά σε στροφές.

Όσον αφορά τους σταθμούς εκπομπής, σημαντική αλλά και μεγάλη επέκταση αποτελεί η τροποποίηση του εργαλείου ώστε να λαμβάνονται υπόψη τα εμπόδια που βρίσκονται ανάμεσα στα οχήματα και τους σταθμούς εκπομπής. Όπως είναι γνωστό, τα εμπόδια μειώνουν λόγο σήματος προς θόρυβο (signal to noise ratio) και δυσχεραίνουν την επικοινωνία των οχημάτων με τους σταθμούς εκπομπής. Με την κατάλληλη τροποποίηση, τα εμπόδια ανάμεσα στα οχήματα και στους σταθμούς εκπομπής θα λαμβάνονται υπόψη, και έτσι, όταν ένα όχημα επιθυμεί την επικοινωνία με κάποιον σταθμό εκπομπής για μεταφόρτωση δεδομένων, θα δίνεται μεγαλύτερη προτεραιότητα σε χωροpixel που διαθέτουν μεγαλύτερο λόγο σήματος προς θόρυβο.

Τέλος, σε αυτήν την κατηγορία βελτιώσεων ανήκει και η υποστήριξη περισσότερων παραμέτρων του περιβάλλοντος και τύπων οχημάτων. Εκτός από ελεύθερα χωροpixel (προσβάσιμος χώρος) και εμπόδια (μη προσβάσιμος χώρος) θα μπορούν να προστεθούν επιπλέον περιοχές, όπως θάλασσα, λάσπη, χώμα κ.α.. Έτσι, το μονο-



Σχήμα 5.1: Ο κύκλος γύρω από το χωροpixel BS υποδηλώνει τον σταθμό εκπομπής. Τα γκρι χωροpixel υποδηλώνουν περιοχές με υψηλό λόγο σήματος προς θόρυβο. Το γαλάζιο χωροpixel υποδηλώνει την περιοχή με μέτριο λόγο σήματος προς θόρυβο, και τέλος, το σκούρο μπλέ χωροpixel υποδηλώνει περιοχές με χαμηλό λόγο σήματος προς θόρυβο.

πάτι κάλυψης που θα παράγετε για κάθε όχημα θα προσαρμόζεται στις δυνατότητές του και στις συνθήκες του περιβάλλοντος. Για παράδειγμα, οχήματα θα υπάρχουν οχήματα με δυνατότητα πλοήγησης μόνο σε περιβάλλοντα θάλασσας αλλά όχι λάσπης και οχήματα με δυνατότητα προσπέλασης μόνο σε χώμα αλλά όχι λάσπη. Έτσι, το εργαλείο θα γίνει πιο ρεαλιστικό και θα αποκτήσει ακόμα περισσότερες πρακτικές εφαρμογές.

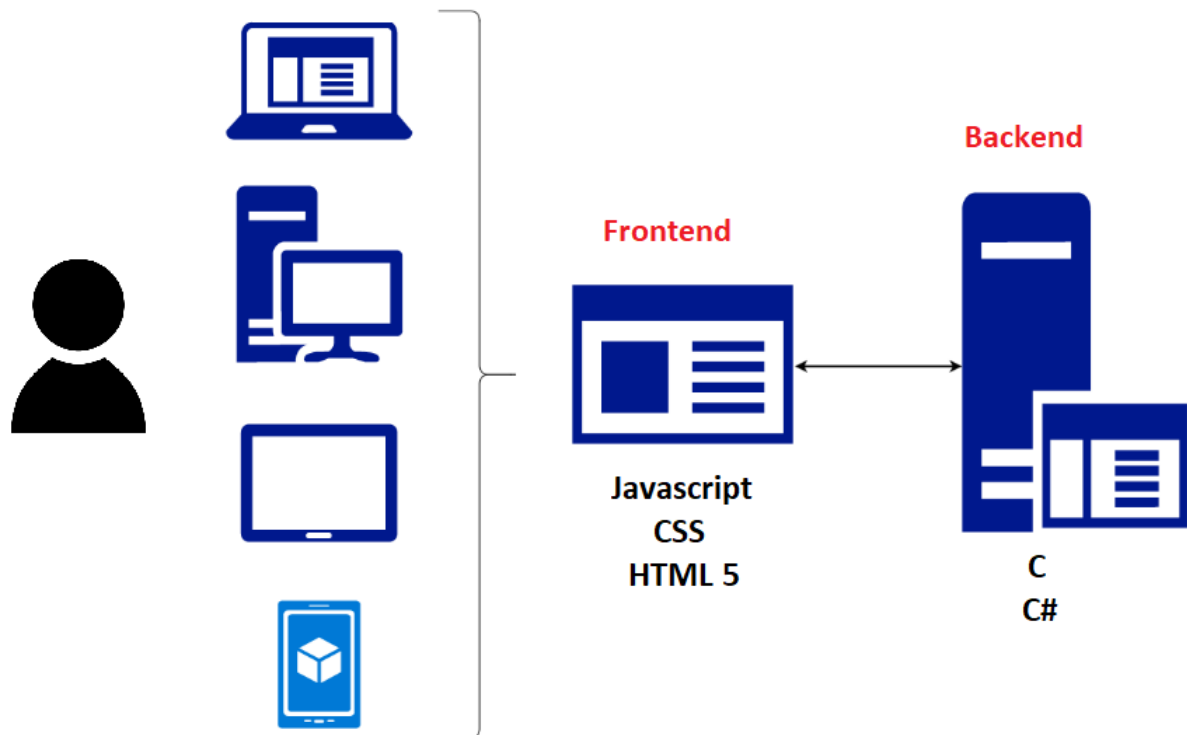
5.3.2 Βελτίωση της απόδοσης του αλγορίθμου

Η αύξηση της απόδοσης αποτελεί μία βελτίωση που θα συμβάλει στην προώθηση της χρήσης του εργαλείου. Η προσαρμογή όλων των αλγορίθμων που χρησιμοποιούνται ώστε να λειτουργούν ανεξάρτητα ο ένας από τον άλλο, θα οδηγήσει στην παραλληλοποίηση του και θα μειώσει τον χρόνο που απαιτείται για την εύρεση του μονοπατιού κάλυψης χώρου. Η βελτίωση αυτή δεν μπορεί να υλοποιηθεί παντού, καθώς υπάρχουν αλγόριθμοι των οποίων οι είσοδοι εξαρτώνται από τις εξόδους

άλλων αλγορίθμων. Σε αυτές τις περιπτώσεις, η βελτίωση μπορεί να γίνει με την μετατροπή των αλγορίθμων σε γλώσσες χαμηλότερου επιπέδου, όπως η C.

5.3.3 Ανάπτυξη διαδικτυακής εφαρμογής

Μία ακόμη σημαντική βελτίωση που θα συμβάλει στην προώθηση της χρήσης του εργαλείου είναι η ανάπτυξη διαδικτυακής εφαρμογής που θα προσφέρει δικτυακά τις υπηρεσίες του εργαλείου. Έτσι, οι χρήστες θα μπορούν να χρησιμοποιούν το εργαλείο χωρίς να τους απασχολεί η εγκατάσταση και η συντήρησή του. Επιπλέον, χρήστες που δεν έχουν στην διάθεσή τους υπολογιστή υψηλών επιδόσεων θα μπορούν να χρησιμοποιούν και να απολαμβάνουν όλες τις δυνατότητες που προσφέρονται. Στο σχήμα 5.2 βλέπουμε την αναπαράσταση της διαδικτυακής εφαρμογής.



Σχήμα 5.2: Αναπαράσταση μοντέλου διαδικτυακής εφαρμογής.

5.3.4 Ανάπτυξη Διεπαφής Προγραμματισμού Εφαρμογών (API)

Τέλος, η ανάπτυξη ενός web API (Application Programming Interface) μπορεί να συνεισφέρει στην διάδοση και γνωστοποίηση του εργαλείου. Χρησιμοποιώντας το API, αυτόνομα οχήματα θα μπορούν να επικοινωνούν απευθείας με τον διακομιστή στον οποίο θα φιλοξενείται το εργαλείο και θα του αποστέλλουν πληροφορίες

σχετικά με την μορφολογία των οχημάτων και του περιβαλλοντος. Ο διακομιστής θα επιστρέφει το μονοπάτι κάλυψης πίσω στα οχήματα, τα οποία στη συνέχεια θα πλοηγούνται στο χώρο.

Παραρτήματα

Παράρτημα Α΄

Εγκατάσταση και χρήση εργαλείου

Το εργαλείο που αναπτύχθηκε υποστηρίζει το Λειτουργικό Σύστημα Microsoft Windows και διατίθεται σε δυαδική εκτελέσιμη μορφή (binary executable file) .exe. Είναι απαραίτητο, τόσο το εκτελέσιμο αρχείο όσο και οι βιβλιοθήκες (.dll) που συμπεριλαμβάνονται να βρίσκονται σε περιοχές καταλόγου που ο χρήστης έχει δικαιώματα ανάγνωσης και εγγραφής. Διαφορετικά, δεν θα μπορέσει να λειτουργήσει σωστά το εργαλείο. Σε παλαιότερες εκδόσεις των Microsoft Windows (Windows XP SP3 και Windows Vista) είναι απαραίτητη η εκτέλεση του εργαλείου ως διαχειριστής. Διαφορετικά, ενδεχεται να παρουσιαστούν σφάλματα κατά την εκτέλεση. Τέλος, όλα τα αρχεία (εκτελέσιμα .exe, βιβλιοθήκες .dll, αρχεία περιβάλλοντος .txt) είναι απαραίτητο να βρίσκονται στον ίδιο κατάλογο.

Βιβλιογραφία

- [1] C Sharp Documentation. <https://docs.microsoft.com/en-us/dotnet/csharp/>. Accessed: 2019-02-27.
- [2] Microsoft .NET Core Guide. <https://docs.microsoft.com/en-us/dotnet/core/>. Accessed: 2019-02-27.
- [3] Xamarin documentation. <https://docs.microsoft.com/en-us/xamarin/>. Accessed: 2019-02-27.
- [4] Abdeldime Abdelgader. Drones and its application in agriculture, 04 2015.
- [5] Héctor Azpúrua, Paulo Rezeck, Douglas Macharet, and Mario Campos. Multi-robot 3d coverage path planning for first responders teams. pages 1374–1379, 08 2016.
- [6] Keshav Bimbraw. Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings*, 1:191–198, 01 2015.
- [7] Vincent Boulanin and Maaike Verbruggen. *Availability and military use of UAVs*, pages 121–132. 08 2017.
- [8] Ching-Yao Chan. Advancements, prospects, and impacts of automated driving systems. *International Journal of Transportation Science and Technology*, 6(3):208 – 216, 2017. Safer Road Infrastructure and Operation Management.
- [9] Howie Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, Oct 2001.
- [10] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon decomposition. In *International Conference on Field and Service Robotics*, January 1997.
- [11] Pamela Das and Richard Horton. Pollution, health, and the planet: Time for decisive action. *The Lancet*, 391, 10 2017.
- [12] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1(1):269–271, December 1959.
- [13] Prof. Amakievi Gabriel. *The History of Science And Technology in Perspective*. 01 2009.

-
- [14] Yoav Gabriely and Elon Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31(1):77–98, Oct 2001.
- [15] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258 – 1276, 2013.
- [16] Quentin Galvane. Automated cinematography with unmanned aerial vehicles. *Eurographics Workshop on Intelligent Cinematography and Editing*, pages 23–30, 01 2016.
- [17] Masoud Gheisari and Behzad Esmaeili. Unmanned aerial systems (uas) for construction safety applications. pages 2642–2650, 05 2016.
- [18] R.L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7:43–57, 02 1985.
- [19] Malcolm J. Zapata, William J. Haynes, Nicholas Kannen, Megan Sullivan, and James Conrad. An autonomous vehicle for space exploration. pages 15–20, 05 2008.
- [20] Aubrey John Russell. Android drone: Remote quadcopter control with a phone. 12 2016.
- [21] Dušan Jovanović, Miro Govedarica, and Dejan Rašić. Remote sensing as a trend in agriculture. *Research Journal of Agricultural Science*, 46:32–37, 09 2014.
- [22] Athanasios Ch Kapoutsis, Savvas A Chatzichristofis, and Elias B Kosmatopoulos. Darp: Divide areas algorithm for optimal multi-robot coverage path planning. *Journal of Intelligent & Robotic Systems*, 86(3):663–680, 2017.
- [23] Kostas Karantininis. *Overview of Greek Agriculture*, pages 5–11. 08 2017.
- [24] Piotr Kardasz and Jacek Doscocz. Drones and possibilities of their using. *Journal of Civil and Environmental Engineering*, 6, 01 2016.
- [25] Muhammad Khan, Bilal A. Alvi, Engr Safi, and Inam Khan. Drones for good in smart cities:a review. 01 2018.
- [26] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.
- [27] Jean-Claude Latombe. *Approximate Cell Decomposition*, pages 248–294. Springer US, Boston, MA, 1991.
- [28] Martin Mares. The saga of minimum spanning trees. *Computer Science Review*, 2:165–221, 12 2008.
- [29] Rolf Möhring, Heiko Schilling, Birk Schütz, Dorothea Wagner, and Thomas Willhalm. Partitioning graphs to speed up dijkstra’s algorithm. volume 3503, pages 189–202, 05 2005.
- [30] Thomas Misa. *History of Technology*, volume 30, pages 5 – 17. 04 2009.

-
- [31] Krishn Mishra, A K Misra, Peter Mueller, Gregorio Martinez Perez, Sanjiv Bhatia, and Yong Wang. Recent advancements in computer and software technology. *TheScientificWorldJournal*, 2014:609512, 06 2014.
- [32] Mohd Shahrieel Mohd Aras, Hyreil Kasdirin, Muhammad Jamaluddin, and Md Basar. Design and development of an autonomous underwater vehicle (auv-fkeutem). *Proceedings of MUCEET2009 Malaysian Technical Universities Conference on Engineering and Technology, MUCEET2009, MS Garden, Kuantan, Pahang, Malaysia*, 01 2009.
- [33] Haibo Niu, S Adams, Kenneth Lee, T Husain, and Neil Bose. Applications of autonomous underwater vehicles in offshore petroleum industry environmental effects monitoring. *Journal of Canadian Petroleum Technology*, 48:12–16, 05 2009.
- [34] Gonzalo Pajares. Advances in sensors applied to agriculture and forestry. *Sensors (Basel, Switzerland)*, 11:8930–2, 12 2011.
- [35] Chintan Pathak and Dr. Hiren Mandalia. Impact of environmental pollution on human future. *World Journal of Environmental Pollution (WJEP)*, 1:8–10, 01 2011.
- [36] Wen-bao Qiao and Jean-Charles Créput. Gpu implementation of borůvka’s algorithm to euclidean minimum spanning tree based on elias method. *Applied Soft Computing*, 76, 12 2018.
- [37] Bharat Rao, Ashwin Goutham Gopi, and Romana Maione. The societal impact of commercial drones. *Technology in Society*, 45:83–90, 05 2016.
- [38] Abdul Rehman, Luan Jingdong, Rafia Khatoon, and M I Hussain. Modern agricultural technology adoption its importance, role and usage for the improvement of agriculture. *American-Eurasian Journal of Agricultural & Environmental Sciences*, 16:284–288, 02 2016.
- [39] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *J. ACM*, 13(4):471–494, October 1966.
- [40] F. Rubin. The lee path connection algorithm. *IEEE Transactions on Computers*, C-23:907–914, 1974.
- [41] Çağdaş Sağır, Günseli Erdem, Bedri Kurtulus, Özgür Avşar, Dilek Kurtulus, Evren Tunca, Mathieu Le Coz, and Moumtaz Razack. Potential of unmanned aerial vehicle (uav) remote sensing in karst springs: Muğla, turkey. 12 2017.
- [42] Antonis Tzounis, Nikolaos Katsoulas, Thomas Bartzanas, and Constantinos Kittas. Internet of things in agriculture, recent advances and future challenges. *Biosystems Engineering*, 164:31–48, 12 2017.
- [43] Hoang Huu Viet, Viet-Hung Dang, Md Nasir Uddin Laskar, and TaeChoong Chung. Ba*: an online complete coverage algorithm for cleaning robots. *Applied Intelligence*, 39(2):217–235, Sep 2013.

-
- [44] Arnold W. Schumann. Precise placement and variable rate fertilizer application technologies for horticultural crops. *HortTechnology*, 20, 02 2010.
- [45] Marten Waanders. Coverage path planning for mobile cleaning robots. 2011.
- [46] Richard Welch, Daniel Limonadi, and Robert Manning. Systems engineering the curiosity rover: A retrospective. pages 70–75, 06 2013.
- [47] Kesheng Wu, Ekow Otoo, and Kenji Suzuki. Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis & Applications*, 12:117–135, 06 2009.
- [48] Zhengwei Yang, Wen-bin WU, Liping Di, and Burak Üstündağ. Remote sensing for agricultural applications. *Journal of Integrative Agriculture*, 16:239–241, 02 2017.