ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

[PLH-311]-Τεχνητή Νοημοσύνη
Αναφορά 1ης προγραμματιστικής άσκησης

Ομάδα χρηστών 28:
Ιωάννης Περίδης
A.M. 2018030069
Γεώργιος Σκουλάς
A.M. 2018030148

Διδάσκων:
Γεώργιος Χαλκιαδάκης

30 Απριλίου 2022

1 Εισαγωγή & Σκοπός:

Σκοπός της εργαστηριακής αυτής άσκησης, ήταν η υλοποίηση και η μελέτη του προβλήματος εύρεσης βέλτιστης διαδρομής με ένα όχημα, μέσω του συντομότερου μονοπατιού, μέχρι να φτάσει στον τελικό του στόχο. Η διαδικασία εύρεσης γίνεται offline, δηλαδή πριν αναχωρήσει το όχημα, επομένως, για να έχουμε γρήγορη ανταπόκριση πρέπει να ελεγχθούν πολλά διαφορετικά μονοπάτια.

Αρχικά, υλοποιήθηκαν οι αλγόριθμοι Weighted A* και IDA* και ύστερα έγινε έλεγχος των αποτελεσμάτων με την χρήση διάφορων ευρετικών οι οποίες και συγκρίθηκαν μεταξύ τους ως προς την αποτελεσματικότητα.

2 Ανάλυση Αλγόριθμων:

Για την υλοποίηση των δύο αλγορίθμων αυτών, χρειάζεται να διατηρηθεί η σειρά με την οποία θα διασχίζονται όλοι οι κόμβοι. Συγκεκριμένα, χρησιμοποιήθηκε η δοθήσα δομή σειράς προτεραιότητας ,Priority Queue (βρίσκεται εντός της βιβλιοθήκης, στο μοντέλο SMP.motion_planner.queue). Αναλυτικά, η λειτουργία της προσθήκης κόμβων έχει ως εξής. Δίνονται σαν ορίσματα οι κόμβοι που θα προστεθούν και η τιμή προτεραιότητάς τους, η οποία δεν είναι άλλη από την τιμή επιστροφής της συνάρτησης αξιολόγισης κόμβου. Επομένως, κάθε φορά που γίνεται μια αφαίρεση από την Priority Queue, παίρνουμε τον πιο φθηνό κόμβο, δηλαδή αυτόν με την μικρότερη τιμή αξιολόγισης.

Στο σημείο αυτό, πρέπει να αναφαιρθεί πως ο λόγος που δεν χρειαστήκε να υπάρχεχι μνήμη των κόμβων που έχουμε επισκευθεί ήδη, είναι πως η εξερεύνηση των κόμβων είναι μονόπλευρη και οι μεταβάσεις συνεχής, άρα έτσι κι αλλιώς δεν θα υπάρξουν επαναλήψεις καταστάσεων.

2.A) Weighted A*:

Έστω, ο συνολικός αριθμός κόμβων που εξερευνήθηκαν n.Τότε έχω αριθμό βηματών, δηλαδή αριθμό κόμβων που εξερευνούνται μετά τον αρχικό ίσο με steps= n-1 (αφού με το πρώτο βήμα ελέγχεται και ο πρώτος κόμβος).

Αρχικά, τοποθετείται σητν ουρά ο πρώτος κόμβος που έχει δοθεί σαν όρισμα και αρχικοποιείται ο μετρητής steps στο 0. Ύστερα, μπαίνουμε σε ένα while loop που τρέχει για όλους τους κόμβους της ουράς την βασική λειτουργία του προγράμματος. Σε κάθε μια επανάληψη, γίνεται pop ο κόμβος με την μικρότερη τιμή και εξερευνόνται όλες οι δυνατές μεταβάσεις του , με ητν χρήση ενός for loop. Με μια if, γίνεται έλεγχος για το αν κάποιο από τα παιδία του κόμβου προκαλέι σύγκρουση , η οποία ανιχνεύεται από την δοθήσα take_step. Στην περίπτωση σύγκρουσης, αγνοούμε το παιδί και προχωράμε στο επόμενο,αλλιώς, ελέγχουμε αν η μετάβαση στο συγκεκριμένο παιδί, δημιουργέι κάποιο μονοπάτι που περνάει από τον στόχο. Αν βρεθέι ένα τέτοιο μονοπάτι, η διαδικασία ολοκληρώνει επιτυχώς, αλλιώς αν δεν βρεθεί στόχος από το παιδί , μπαίνει στην ουρά για μεταγενέστερη εξερεύνηση όταν φτάσει η σειρά του.

Η διαφορά του αλγορίθμου Weighted A* με τον απλό A*, είναι πώς τώρα τοποθετείται ένα βάρος w στην ευρετική , στον υπολογισμό της αξιολόγισης. Ανάλογα το βάρος αυτό, θα έχουμε και διαφορές μεταξύ των αλγορίθμων.

2.B) IDA*:

Αντίστοιχα και ο αλγόριθμος IDA*, είναι μια παραλλαγή του αλγορίθμου A* με κάποιους περιορισμούς. Αρχικά, ορίζεται ένα όριο ίσο με την τιμή αξιολόγισης της συνάρτησης του πρώτου κόμβου. Η διαφορά τους λοιπόν είναι πως στον IDA*, εξερευνόνται μόνο οι κόμβοι , οι οποίοι έχουν τιμή συνάρτησης αξιολόγισης μικρότερη ή ίση από το τρέχον όριο (το όριο αλλάζει).

Η λειτουργία του αλγορίθμου, είναι ίδια με αυτήν του A*, αλλά εκτελείται μέχρι η τιμή αξιολόγισης του φθινότερου κόμβου της ουράς, να ξεπεράσει το όριο. Έπειτα, το όριο πρέπει να αναναιωθεί (αυξηθεί) με νέα τιμή την τιμή της αξιολόγισης του κόμβου που παραβίασε πρώτος το όριο. Ακόμη, πρέπει η ουρά αναμονής (δηλαδή το fringe) να τεθεί ξανά στην αρχική του κατάσταση , δηλαδή περιέχοντας μόνο τον αρχικό κόμβο, έτσι ώστε να ξαναρχίσει η λειτουργία του A* από την αρχή και μέχρι την επόμενη παραβίαση ορίου.

Είναι φανερό, πως ο IDA*, σε σχέση με τον A*, εξοικονομεί χώρο , δηλαδή μνήμη, εφόσον αποθηκεύει λιγότερη πληροφορία την οποία αφαιρεί κι όλας περιοδικά. Παρόλα αυτά, είναι λογικό να μας κοστίσει περισσότερο χρονικά , λόγω των πολλαπλών επαναλήψεων που συμβαίνουν.

3 Επιλογή Ευρετικών:

Για τα πειράματα και την σύγκριση των αποτελεσμάτων, χρησιμοποιήθηκαν δύο διαφορετικές ευρεστικές. Η Ευκλείδεια απόσταση (Euclidian distance) και η απόσταση Μανχάταν (Manhattan distance).

3.A) Euclidian Distance:

Η Ευκλείδεια απόσταση, είναι μια αρκετά καλή και παραδεκτή λύση για απλά προβλήματα εύρεσης συντομότερου μονοπατιού και λειτουργεί ως εξής. Υπολογίζει την ευθεία απόσταση από το σημείο που εξετάζεται στο σημείο στόχο , δηλαδή βρίσκει την ελάχιστη δυνατή απόσταση μεταξύ τους. Ακόμη, δεν έχει περιορισμούς κινήσεων σε συγκεκριμένους άξονες (σαν πλέγμα), αλλά έχει την ελευθερεία να κινηθείς και με γωνίες. Τέλος, είναι η φθηνότερη εκτίμηση που μπορεί να βρεθεί.

Παρόλα αυτά, αυτό σημαίνει πως υπάρχει δυνατότητα βελτίωσης και εύρεσης ευρεστικών οι οποίες θα είναι καλύτερες και θα έχουν και μικρότερο branching factor (παράγωντας διακλαδώσεων). Ένα παραπάνω πρόβλημα που δημιουργείται στην δική μας περίπτωση είναι πως για να γίνει μια ευθεία κίνηση, θα χρειαστούν πολλαπλές αποφυγές εμποδίων που θα βρίσκονται στην άμεση αυτή διαδρομή μεταξύ του κόμβου και του στόχου.

Λόγω αυτών, οδηγηθήκαμε στην παρακάτω ευρετική, η οποία είναι επίσης παραδεκτή καθώς και δεν ξεπερνά το τελικό κόστος.

3.B) Manhattan Distance:

Η απόσταση Μανχάταν, επιλέχθηκε διότι είναι μια ευρετική που επιτρέπει κινήσεις που μπορούν να οδηγήσουν το όχημα πάνω και μετά δεξιά , όπως και αντίστοιχα κάτω και μετά δεξιά. Αποτέλεσμα αυτού είναι πως δίνει μια πολύ καλύτερη συμπεριφορά από πριν στο πρόβλημα της αποφυγής εμποδίων. Αυτό διότι για να αποφευχθούν τα εμπόδια θα πρέπει να κινηθεί το όχημα όπως προαναφαίρθηκε ,αντιμετωπίζοντας όλα τα εμπόδια σαν blocks στον χώρο.

4 Σύγκριση & Σχολιασμός:

Για να γίνει σωστά η σύγκριση και κατανόηση των αποτελεσμάτων δημιουργήθηκαν τα παρακάτω πινακάκια που μας δείχνουν πόσους κόμβους επισκεπτόταν κάθε αλγόριθμος σε όλα τα σενάρια και για όλα τα βάρη που χρησιμοποιήθηκαν:

Scenario 1					
A*			**IDA***		
	Euclidean	**Manhattan**		**Euclidean**	**Manhattan**
w=0			**w=0**		
w=1	53	42	**w=1**	432	272
w=2	30	18	**w=2**	30	18
w=3	30	18	**w=3**	30	18

Πίνακας 1

Scenario 2					
A*			**IDA***		
	Euclidean	**Manhattan**		**Euclidean**	**Manhattan**
w=0			**w=0**		
w=1	385	246	**w=1**	4530	3592
w=2	154	89	**w=2**	154	89
w=3	198	96	**w=3**	198	96

Πίνακας 2

Scenario 3					
A*			IDA*		
	Euclidean	Manhattan		Euclidean	Manhattan
w=0			w=0		
w=1	398	274	w=1	8814	2367
w=2	425	95	w=2	425	359
w=3	425	95	w=3	454	163

Πινακας 3

	Euclidean		Manhattan	
	heuristic	estimated	heuristic	estimated
Scenario 1	30	35.14	32.56	35.32
Scenario 2	50	55.33	52.56	55.51
Scenario 3	55	69.08	59,56	69,08

Πίνακας 4

4.Α) Σύγκριση A*-IDA*:

Αρχικά, είναι αναγκαίο να αναφερθεί ότι στους 2 αλγόριθμους που υλοποιήθηκαν παρατηρήθηκαν αρκετές ομοιότητες, αλλά και σημαντικές διαφορές ως προς τον τρόπο που υπολογίζεται το μικρότερο μονοπάτι. Πιο συγκεκριμένα παρατηρείται από τους πινακες 1,2,3 ότι ο A* αλγόριθμος επισκέπτεται πολύ λιγότερους κόμβους για να βρει το shortest path. Αυτό συμβαίνει διότι ο IDA* όταν ξεπεράσει το όριο που έχει οριστεί αρχίζει την διαδικασία αναζήτησης από την αρχή με καινούριο όριο. Συνεπώς, αν αυτοσκοπός μας είναι η ταχύτερη εύρεση του συντομότερου μονοπατιού offline χωρίς να έχουμε περιορισμό μνήμης τότε θα προτιμηθεί ο αλγόριθμος A*.

Αν όμως στις προδιαγραφές του προβλήματος που μας έχει δοθεί έχουμε περιορισμό μνήμης και ο A* δεν μας καλύπτει τότε θα χρησιμοποιηθεί ο αλγόριθμος του IDA*. Όπως έχει προαναφερθεί ο αλγόριθμος αυτός βολεύει στα προβλήματα στα οποία υπάρχει περιορισμός μνήμης, καθώς υπάρχει ένα όριο κόστους και κάθε φορά που δεν υπάρχουν στο fringe κόμβοι που να το ικανοποιούν τότε η αναζήτηση αρχίζει από την αρχή με νέο όριο. Ακόμα, είναι αναγκαίο να αναφερθεί ότι ο IDA* χρησιμοποιεί ένα από τα βασικά χαρακτηριστικά του A* για εξερεύνηση πρώτα των κόμβων με μικρότερο κόστος, χρησιμοποιώντας την συνάρτηση αξιολόγησης του A*, και το χαρακτηριστικό της επαναλαμβανόμενης εμβάθυνσης του DFS. Συμπερασματικά, ο A* παρουσιάζει καλύτερη απόδοση χρονικά, ενώ ο IDA* χωρικά.

Τέλος, κατά την διάρκεια των πειραματισμων που έγιναν για την τιμή του w παρατηρήθηκε ότι για μεγάλο w ο αλγόριθμος IDA* τείνει να γίνει σαν τον A*. Αυτό είναι λογικό να συμβεί καθώς κυρίαρχος όρος γίνεται το w*h(n) και το κόστος g(n) θα είναι μηδαμινό με αποτέλεσμα να λαμβάνεται υπόψη στην λήψη απόφασης του νέου κόμβου που θα διερευνηθεί. Σε αυτή λοιπόν την περίπτωση το όριο δε θα ξεπεραστεί ποτέ και έτσι αν δεν έχουμε επανεκκίνηση ο IDA* ταυτίζεται με τον A*.

4.B) Σύγκριση Παραλλαγών Weighted A*

Σε αυτό το μέρος πραγματοποιήθηκαν διάφοροι πειραματισμοί έτσι ώστε να βρεθεί η τιμή για το w, στην οποία θα έχουμε τα καλύτερα αποτελέσματα. Ο τύπος του f(n) είναι:

$$\boldsymbol{f(n) = g(n) + w * h(n)}$$

Κατά την διάρκεια των πειραματισμών ελέγχθηκαν τιμές για το w σε ένα εύρος [0,10] και αυτό που παρατηρήθηκε ήταν ότι από κάποιο w και μετά αύξηση του w δεν επηρεάζει σχεδόν καθόλου το σύνολο των κόμβων που θα επισκεφτει. Αυτό συνέβει για όλα τα σενάρια και συνήθως η παρατήρηση αυτή γινόταν για w ίσο με 3 ή 4. Η παρατήρηση αυτή ήταν απολύτως λογική, καθώς κυρίαρχος όρος στην f(n) γίνεται το wh(n) και το g(n) συνεισφέρει ελάχιστα στην f(n). Για μεγάλες τιμές του w, παρατηρείται ότι ο A* μετατρέπεται σε Greedy Best First Search. Τέλος, για w=0 ο αλγόριθμος μας δεν αξιοποιεί την ευρετική μας συνάρτηση, με αποτέλεσμα ο συνολικός αριθμός των επισκεπτόμενων κόμβων να αυξηθεί.

4.C) Σύγκριση Ευρετικών:

Για την υλοποίηση των 2 αλγορίθμων δημιουργήθηκαν 2 ευρετικές συναρτήσεις. Η 1η ευρετική που υλοποιήθηκε ήταν η ευκλείδεια απόσταση, ενώ η 2η ήταν η απόσταση manhattan.

Αρχικά, βλέποντας τους πινακες 1,2,3 μπορούμε να συμπεράνουμε ότι η ευκλείδεια απόσταση ήταν λιγότερο αποδοτική από την απόσταση manhattan και στα 3 σενάρια. Εν συνεχεία, παρατηρείται από τον πίνακα 4 ότι η ευκλείδεια ευρετική είναι πιο αισιοδοξη από την manhattan, πράγμα το οποίο συνεπάγεται στο να είναι λιγότερο κυρίαρχη η ευκλειδεια. Αυτό σημαίνει ότι θα γίνουν περισσότερες επεκτάσεις κόμβων.

Η δεύτερη ευρετική που υλοποιήθηκε ήταν η απόσταση Manhattan. η ευρετική αυτή όπως παρατηρείται και στα πινακάκια είναι πιο αποδοτική. Η ευρετική αυτή είναι πιο κυρίαρχη σε σχέση με την ευκλείδεια απόσταση. Ακόμα παρατηρήθηκε ότι αυτή η ευρετική ήταν ικανότερη να αποφεύγει εμπόδια γρηγορότερα από ότι της ευκλείδειας.

Συμπερασματικά, μπορούμε να πούμε με ακρίβεια ότι και για τους 2 αλγόριθμους(A*, IDA*) η χρήση της απόστασης Manhattan μας δίνει αποδοτικότερη λύση.