

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΤΝΗ 2022

Εργασία Εξαμήνου

Νικόλαος-Μάριος Κόντος



Εισαγωγή

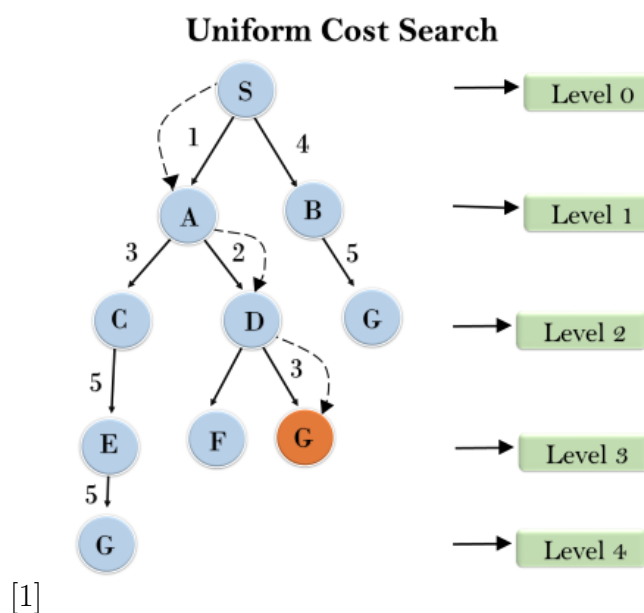
Κατά τη διάρκεια του εξαμήνου μας ανατέθηκε να υλοποιήσουμε 2 αλγορίθμους ευρετικής και τυφλής αναζήτησης, τον A* και ομοιόμορφου κόστους αναζήτησης (UCS).

Οι 2 αυτοί αλγόριθμοι κατά βάση χρησιμοποιούν την αναζήτηση σε πλάτος (BFS) και μοιάζουν σε πολύ μεγάλο βαθμό μεταξύ τους, έχουν όμως ως βασικότερη διαφορά την χρήση της ευρετικής συνάρτησης που αξιοποιεί ο A*. Στην συνέχεια θα γίνει μία σύντομη περιγραφή κάθε αλγορίθμου και τελικά μία σύγκριση των 2 ως προς την απόδοσή τους βάσει του προβλήματος που μας δόθηκε να επιλύσουμε.

UCS

Δοθείσας μίας αρχικής κατάστασης (κόμβου) ο UCS θα βρεί το μονοπάτι με το μικρότερο κόστος από την αρχική κατάσταση στην τελική την οποία έχουμε ορίσει. Σε κάθε βήμα επιλέγεται για επέκταση κόμβος του μετώπου αναζήτησης με το μικρότερο $g(n)$, όπου $g(n)$ είναι το κόστος του μονοπατιού από την ρίζα (Α.Κ.) μέχρι τον κόμβο n .

Στην παρακάτω εικόνα φαίνεται ο τρόπος λειτουργίας από την Α.Κ. S στην τελική κατάσταση G . Να σημειωθεί ότι ο BFS είναι μια περίπτωση του UCS όπου τα κόστη είναι όλα ίσα.



[1]

Πολυπλοκότητα Χρόνου & Χώρου

Ορίζω C^* το κόστος της βέλτιστης λύσης και ϵ το κάθε βήμα που μας πηγαίνει πιο κοντά στην Τ.Κ. Τότε ο αριθμός των βημάτων είναι $C^*/\epsilon + 1$.

Επομένως στην χειρίστη περίπτωση η ομοιόμορφου κόστους αναζήτησης είναι πολυπλοκότητας $O(b^{1+\lceil C^*/\epsilon \rceil})$

Η ίδια λογική ακριβώς ισχύει και για την πολυπλοκότητα χώρου με χειρίστη περίπτωση $O(b^{1+\lceil C^*/\epsilon \rceil})$

Επομένως η UCS είναι εκθετικής πολυπλοκότητας χώρου και χρόνου. Είναι πλήρης και βέλτιστη διότι αν υπάρχει μονοπάτι θα επιστρέψει αυτό με το μικρότερο κόστος.

A*

Ο A* (A-star) ^[2] ή Άλφα-άστρο είναι ένας αλγόριθμος διάσχισης γράφων και αναζήτησης μονοπατιών. Λειτουργεί όμοια με τον UCS, όμως έχει την βασική διαφορά στον τρόπο επιλογής του κόμβου προς επέκταση. Προς επέκταση επιλέγεται ο κόμβος που έχει το μικρότερο $e(n) = g(n) + h(n)$.

$g(n)$ είναι η πραγματική απόσταση του αρχικού κόμβου από κάποιο κόμβο n

$h(n)$ είναι η ερετική συνάρτηση που δίνει μια εκτίμηση της απόστασης n από την Τ.Κ. (π.χ. Ευκλείδεια μετρική ή απόσταση Μανχάταν)

Αν η ευρετική συνάρτηση είναι αποδεκτή τότε ο A* είναι βέλτιστος και πλήρης. Για να είναι αποδεκτή η ευρετική συνάρτηση θα πρέπει η πραγματική απόσταση κάποιας κατάστασης n από την πλησιέστερη Τ.Κ. $a(n)$ να είναι μεγαλύτερη ή τουλάχιστον ίση με την τιμή της ευρετικής συνάρτησης $h(n)$

Για να είναι η $h(n)$ αποδεκτή θα πρέπει $h(n) \leq a(n), \forall n$

Σύγκριση & Συμπεράσματα

Παρουσίαση του προβλήματος

Έχουμε έναν λαβύρινθο μεγέθους $N \times N$ με κάποια εμπόδια. Θέλουμε από μία αρχική κατάσταση G το ελάχιστο μονοπάτι προς τη πλησιέστερη τελική $S1$ ή $S2$. Το μέγεθος του λαβυρίνθου, η αρχική και τελική κατάσταση όπως και η πιθανότητα να υπάρχει εμπόδιο στο λαβύρινθο δίνονται από τον χρήστη.

Οι επιτρεπτές κινήσεις από ένα κελί (x, y) σε ένα άλλο γειτονικό (x', y') είναι οριζόντια, κάθετα και διαγώνια, ορίζουμε ως $\Delta = |val(x, y) - val(x', y')|$

. Το κόστος της διαγώνιας κίνησης είναι $\Delta + 0.5$ και της κάθετης ή οριζόντιας $\Delta + 1$.

1	1	2	3	1	3
X	2	G	X 2	1	2
1	X	X	3	2	3
2	S2	1	1	1	1
2	1	3	S1	3	2
3	2	X	1	1	3

Παράδειγμα ενός λαβυρίνθου μεγέθους 6×6

Ευρετική Συνάρτηση

Για την κατασκευή της ευρετικής συνάρτησης θα χαλαρώσουμε το πρόβλημα και θα κατασκευάσουμε μια συνάρτηση στο χαλαρωμένο πρόβλημα.

Έστω ότι όλα τα Δ έχουν ίδια τιμή, δηλαδή όλα τα $val(x, y)$ είναι ίσα. Τότε το ελάχιστο μονοπάτι που θα μπορούσε να υπάρξει σε αυτό τον λαβύρινθο είναι ένα μονοπάτι που έχει μόνο διαγώνια κατεύθυνση. Το σημαντικό είναι πόσες διαγώνιες θέσεις απέχει ο τωρινός κόμβος από την TK. Θα θέλαμε μια διαγώνια κίνηση να έχει το πολύ ευρετικό κόστος 0.5. Θέτω ως $\Delta X = |x - x'|$ και $\Delta Y = |y - y'|$. Η ευρετική συνάρτηση η που υλοποιήσαμε είναι η $h = \Delta X * 0.25 + \Delta Y * 0.25$ η οποία είναι αποδεκτή διότι δεν θα ξεπεράσει ποτέ το πραγματικό κόστος του κόμβου προς την TK.

Συμπεράσματα

Τρέχοντας τους αλγόριθμους για τον ίδιο λαβύρινθο για διάφορες τιμές του N και ρ έγιναν κάποιες παρατηρήσεις. Σε γενικές γραμμές ο A^* είναι ο πιο γρήγορος αλγόριθμος από τους 2 και αυτό γιατί χρησιμοποιεί την ευρετική συνάρτηση που του επιτρέπει να έχει μια αίσθηση της κατεύθυνσης προς την οποία ψάχνει. Επιπλέον παρατηρήθηκε ότι ο A^* χρειαζόταν σχεδόν τις μισές επεκτάσεις απότι ο UCS και ο χρόνος εκτέλεσης

του ήταν περίπου τα μισά ms.

Όσο το N αυξανόταν τόσο αυξανόταν ο χρόνος εκτέλεσης των 2 αλγορίθμων καθώς ήταν πολύ περισσότεροι οι κόμβοι που έπρεπε να ερευνηθούν. Όσο πιο πολλά μπλοκαρισμένα κελιά είχαμε τόσο λιγότερο χρόνο χρειαζόταν ο αλγόριθμος διότι δεν χρειαζόταν να γίνει κάποιος υπολογισμός για αυτά τα κελιά.

References

- [1] <https://www.javatpoint.com/ai-uninformed-search-algorithms>
- [2] https://en.wikipedia.org/wiki/A*_search_algorithm