

Τεχνητή Νοημοσύνη

Κυριάκος Λάμπρος Κιουράνας

AM: 1115201900238

Πρόβλημα 2

Σε αυτό το πρόβλημα, εξετάζουμε τις τρεις διαφορετικές στρατηγικές αναζήτησης (BFS, DFS, IDS) σε ένα δέντρο με παράγοντα διακλάδωσης $b = 3$ και στόχο σε βάθος $d = 4$.

Ας συνοψίσουμε τα αποτελέσματα που βρήκαμε:

1. **Αναζήτηση Πρώτα Κατά Πλάτος (BFS):** Επεκτείνει όλους τους κόμβους μέχρι το βάθος 4, δηλαδή 121 κόμβους.
2. **Αναζήτηση Πρώτα Κατά Βάθος (DFS):**
 - **Καλύτερη Περίπτωση:** Εάν ο στόχος είναι ο πρώτος κόμβος που θα συναντηθεί στο βάθος 4, η DFS επεκτείνει μόνο 5 κόμβους.
 - **Χειρότερη Περίπτωση:** Εάν ο στόχος είναι ο τελευταίος κόμβος ή δεν υπάρχει, τότε επεκτείνονται 121 κόμβοι (όλοι οι κόμβοι μέχρι το βάθος 4), όπως στο BFS.
3. **Αναζήτηση με Επαναληπτική Εκβάθυνση (IDS):** Απαιτεί 179 επεκτάσεις κόμβων συνολικά, λόγω των πολλαπλών αναζητήσεων DFS με αυξανόμενο βάθος.

Με βάση τα παραπάνω:

- Η **BFS** εγγυάται την εύρεση του στόχου στον μικρότερο δυνατό αριθμό βημάτων (εάν υπάρχει στο συγκεκριμένο βάθος), αλλά επεκτείνει όλους τους κόμβους μέχρι το βάθος του στόχου.
- Η **DFS** είναι πιο αποδοτική στην καλύτερη περίπτωση, όμως μπορεί να καταλήξει σε μεγάλο αριθμό επεκτάσεων στη χειρότερη περίπτωση.
- Η **IDS** παρέχει μια πιο ισορροπημένη λύση, συνδυάζοντας τα πλεονεκτήματα του BFS (εύρεση του στόχου σε ελάχιστο βάθος) και του DFS (μικρή χρήση μνήμης), αλλά επεκτείνει περισσότερους κόμβους λόγω των επαναλήψεων.

Πρόβλημα 3

Για το πρόβλημα εύρεσης βέλτιστης διαδρομής σε ένα πλέγμα 10×10 με τον αλγόριθμο A^* , χρησιμοποιώντας μια παραδεκτή ευρετική συνάρτηση, ακολουθούμε τα εξής βήματα:

1. Ευρετική Συνάρτηση

Η ευρετική συνάρτηση που επιλέχθηκε είναι:

$$h(n) = \frac{\text{Απόσταση Manhattan από τον κόμβο } n \text{ στον στόχο}}{2}$$

Αυτή είναι **παραδεκτή** (admissible) επειδή δεν υπερεκτιμά το πραγματικό κόστος (δηλαδή το κόστος της διαδρομής από το σημείο εκκίνησης ως τον στόχο).

2. Βήματα του Αλγορίθμου A*

- **Αρχικοποίηση:** Ορίζουμε τον κόμβο εκκίνησης “S” με αρχικό κόστος $g(S) = 0$ και υπολογίζουμε $f(S) = g(S) + h(S)$.
- **Επέκταση Κόμβων:** Σε κάθε βήμα, επιλέγουμε τον κόμβο με το μικρότερο $f(n)$, όπου $f(n) = g(n) + h(n)$. Εξετάζουμε τους γειτονικούς κόμβους και προσθέτουμε το αντίστοιχο κόστος κίνησης.
- **Εύρεση Διαδρομής:** Συνεχίζουμε την επέκταση μέχρι να φτάσουμε στον στόχο “G” (τερματικό σημείο). Η βέλτιστη διαδρομή είναι αυτή με το μικρότερο συνολικό κόστος.

3. Παραδείγματα Ευρετικών Συναρτήσεων

- **Απόσταση Manhattan:**

$$h_1(n) = \text{Απόσταση Manhattan από } n \text{ στον στόχο}$$

Είναι παραδεκτή, καθώς αποτελεί υποεκτίμηση του κόστους.

- **Μισή Απόσταση Manhattan:**

$$h_2(n) = \frac{\text{Απόσταση Manhattan από } n \text{ στον στόχο}}{2}$$

Επίσης παραδεκτή και ενδέχεται να είναι πιο ακριβής σε πλέγματα με κόστη κίνησης, κατευθύνοντας τον A* πιο κοντά στη βέλτιστη διαδρομή.

4. Ανάλυση με Εναλλακτικές Στρατηγικές Αναζήτησης

- **Αναζήτηση Πρώτα σε Πλάτος (BFS):** Εξετάζει κάθε επίπεδο πλήρως, χωρίς να χρησιμοποιεί ευρετική. Συνεπώς, επεκτείνει περισσότερους κόμβους, ειδικά σε μεγαλύτερα πλέγματα.
- **Αναζήτηση Πρώτα σε Βάθος (DFS):** Πηγαίνει όσο το δυνατόν βαθύτερα πριν επιστρέψει, κάτι που δεν εγγυάται βέλτιστη διαδρομή σε πλέγμα με κόστη.
- **Αναζήτηση με Επαναληπτική Εκβάθυνση (IDS):** Διευρύνει το βάθος διαδοχικά, αλλά δεν αξιοποιεί ευρετική πληροφόρηση, με αποτέλεσμα να επεκτείνει περισσότερους κόμβους σε μεγάλα πλέγματα.

- **Απληστική Αναζήτηση Πρώτα στο Καλύτερο:** Επεκτείνει μόνο με βάση την ευρετική τιμή, κάτι που δεν εγγυάται βέλτιστη διαδρομή.
- **Αναζήτηση A*:** Συνδυάζει το πραγματικό κόστος $g(n)$ και την ευρετική $h(n)$ για πιο αποδοτική εύρεση βέλτιστης διαδρομής με το λιγότερο δυνατό αριθμό επεκτάσεων κόμβων.

Πρόβλημα 4

Στο πρόβλημα αυτό εξετάζουμε την αμφίδρομη αναζήτηση και τις διαφορετικές παραλλαγές της, αξιολογώντας την πληρότητα, τη βέλτιστη απόδοση και τους τρόπους ελέγχου συνάντησης των δύο αναζητήσεων.

Αμφίδρομη Αναζήτηση

Η αμφίδρομη αναζήτηση περιλαμβάνει δύο ταυτόχρονες αναζητήσεις:

- Μία αναζήτηση **προς τα εμπρός** από την αρχική κατάσταση.
- Μία αναζήτηση **προς τα πίσω** από την κατάσταση στόχου.

Στόχος είναι οι δύο αναζητήσεις να συναντηθούν κάπου ενδιάμεσα, μειώνοντας το συνολικό χρόνο αναζήτησης σε σχέση με μια μονοκατευθυντική αναζήτηση.

Ερώτημα (α): BFS προς τα εμπρός και Αναζήτηση Περιορισμένου Βάθους προς τα πίσω

- **Πληρότητα:**
 - Η **BFS** είναι πλήρης, καθώς εξερευνά όλους τους κόμβους σε κάθε επίπεδο.
 - Η **αναζήτηση περιορισμένου βάθους** δεν είναι πλήρης, καθώς μπορεί να αποτύχει να βρει λύσεις που βρίσκονται πέρα από το όριό της.
 - **Συμπέρασμα:** Ο συνδυασμός δεν είναι πλήρης, διότι η αναζήτηση προς τα πίσω μπορεί να μην συναντήσει την αναζήτηση προς τα εμπρός.
- **Βέλτιστοτητα:**
 - Η **BFS** είναι βέλτιστη όταν όλα τα κόστη είναι ίσα.
 - Η **αναζήτηση περιορισμένου βάθους** δεν είναι βέλτιστη.
 - **Συμπέρασμα:** Ο συνδυασμός δεν είναι βέλτιστος.

Ερώτημα (β): IDS προς τα εμπρός και Αναζήτηση Περιορισμένου Βάθους προς τα πίσω

• **Πληρότητα:**

- Η IDS είναι πλήρης, καθώς αυξάνει το όριο βάθους επαναληπτικά.
- Η αναζήτηση περιορισμένου βάθους δεν είναι πλήρης.
- **Συμπέρασμα:** Ο συνδυασμός δεν είναι πλήρης λόγω του περιορισμένου βάθους προς τα πίσω.

• **Βέλτιστοτητα:**

- Η IDS είναι βέλτιστη για ίσα κόστη.
- Η αναζήτηση περιορισμένου βάθους δεν είναι βέλτιστη.
- **Συμπέρασμα:** Ο συνδυασμός δεν είναι βέλτιστος.

Ερώτημα (γ): A* προς τα εμπρός και Αναζήτηση Περιορισμένου Βάθους προς τα πίσω

• **Πληρότητα:**

- Ο A* είναι πλήρης με παραδεκτή ευρετική συνάρτηση.
- Η αναζήτηση περιορισμένου βάθους δεν είναι πλήρης.
- **Συμπέρασμα:** Ο συνδυασμός δεν είναι πλήρης, λόγω της μη πληρότητας της αναζήτησης προς τα πίσω.

• **Βέλτιστοτητα:**

- Ο A* είναι βέλτιστος με παραδεκτή και συνεπή ευρετική.
- Η αναζήτηση περιορισμένου βάθους δεν είναι βέλτιστη.
- **Συμπέρασμα:** Ο συνδυασμός δεν είναι βέλτιστος.

Ερώτημα (δ): A* προς τα εμπρός και A* προς τα πίσω

• **Πληρότητα:**

- Και οι δύο αναζητήσεις A* είναι πλήρεις, με παραδεκτή ευρετική.
- **Συμπέρασμα:** Ο συνδυασμός είναι πλήρης.

• **Βέλτιστοτητα:**

- Ο A* είναι βέλτιστος με παραδεκτή και συνεπή ευρετική.
- **Συμπέρασμα:** Ο συνδυασμός είναι βέλτιστος.

Έλεγχος Συνάντησης των Δύο Αναζητήσεων

Για την αποδοτική ανίχνευση της συνάντησης των δύο αναζητήσεων:

- **Χρησιμοποιούμε σύνολα (sets):** Κρατάμε ένα σύνολο με τους κόμβους που έχει επισκεφτεί η κάθε αναζήτηση.
- **Έλεγχος κατά την επέκταση:** Όταν επεκτείνουμε έναν κόμβο σε μία αναζήτηση, ελέγχουμε αν αυτός ο κόμβος βρίσκεται στο σύνολο της άλλης αναζήτησης.
- **Συνάντηση:** Αν βρεθεί κοινός κόμβος, οι αναζητήσεις συναντήθηκαν και μπορούμε να συνθέσουμε τη διαδρομή.