ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
DEPARTMENT OF INFORMATICS

Τεχνητή Νοημοσύνη

2024–2025

Το πρόβλημα της μετάβασης των κανιβάλων και των ιεραποστόλων

3120009 Ανδρίτσος Γεώργιος

3190209 Ανθίππη Φατσέα

3210031 Πηνελόπη Γιάννου Ρίζου

Περιεχόμενα

Εισαγωγή

Η παρούσα αναφορά εστιάζει στην επίλυση του κλασικού γρίφου των Ιεραποστόλων και των Κανιβάλων, χρησιμοποιώντας τον αλγόριθμο αναζήτησης **A***. Ο στόχος του γρίφου είναι η ασφαλής μεταφορά όλων των ιεραποστόλων και των κανίβαλων από την αριστερή στη δεξιά όχθη ενός ποταμού, μέσω μιας βάρκας με περιορισμένη χωρητικότητα. Οι περιορισμοί του προβλήματος περιλαμβάνουν:

1. Τη διατήρηση της ασφάλειας των ιεραποστόλων, δηλαδή, σε καμία όχθη, ούτε και στην βάρκα, ο αριθμός των κανίβαλων δεν πρέπει να υπερβαίνει αυτόν των ιεραποστόλων.
2. Τη μεταφορά τουλάχιστον ενός και το πολύ **M** ατόμων ανά διαδρομή με τη βάρκα.
3. Τον περιορισμό στις συνολικές επιτρεπόμενες διασχίσεις.

Ο αλγόριθμος **A*** συνδυάζει το κόστος που έχει ήδη διανυθεί **(g)** και μια ευρετική εκτίμηση του υπολειπόμενου κόστους **(h)**, παρέχοντας μια αποτελεσματική και βέλτιστη λύση μέσω της συνάρτησης **f(n)=g(n)+h(n)**.

Ο κώδικας έχει σχεδιαστεί ώστε να είναι ευέλικτος, επιτρέποντας την προσαρμογή κρίσιμων παραμέτρων όπως ο αριθμός των ατόμων και η χωρητικότητα της βάρκας.

State

Η κλάση **State**, αναπαριστά μια κατάσταση του γρίφου των Ιεραποστόλων και των Κανιβάλων. Η κλάση παρέχει τις απαραίτητες λειτουργίες για να μπορέσει να χρησιμοποιηθεί στον αλγόριθμο αναζήτησης **A***.

Ακολουθεί αναλυτική περιγραφή των μεθόδων της κλάσης.

1. Η State αρχικοποιείται με τα εξής χαρακτηριστικά

__init__(self, numder, left_missionaries, left_cannibals, boat_left=True, father=None, g=0, boat_capacity=2)
Number: Ο αρχικός αριθμός του πλήθους των ιεραποστόλων και κανιβάλων που βρίσκονται στην αριστερή όχθη, χρησιμοποιείται για να υπολογιστεί ο αριθμός των ιεραποστόλων και κανιβάλων στην δεξιά όχθη.
left_missionaries και left_cannibals: Το πλήθος των ιεραποστόλων και κανιβάλων που βρίσκονται στην αριστερή όχθη.
boat_left: Boolean που δείχνει αν η βάρκα είναι στην αριστερή όχθη.
father: Η προηγούμενη κατάσταση.
_g: Το κόστος που έχει διανυθεί μέχρι την τρέχουσα κατάσταση (βάθος στο δέντρο αναζήτησης).
boat_capacity: Η μέγιστη χωρητικότητα της βάρκας.
_h: Η ευρετική συνάρτηση (εκτιμώμενο υπόλοιπο κόστος).
_f: Το συνολικό κόστος **f(n)=g(n)+h(n)**.

2. Η ευρετική heuristic(self)

Εκτιμά το κόστος για την μεταφορά όλων των ατόμων, λαμβάνοντας υπόψη το πλήθος τους στην αριστερή όχθη, τη χωρητικότητα της βάρκας και την τρέχουσα θέση της. Για τη μεταφορά όλων των ανθρώπων, ο ελάχιστος αριθμός διαδρομών υπολογίζεται ως:

$$trips = \lceil (people\_left) / (boat\_capacity - 1) \rceil$$

people_left = άτομα που είναι στην αριστερή όχθη.
boat_capacity – 1 = η χωρητικότητα της βάρκας -1 για την θέση του οδηγού.

Εάν υπάρχουν περισσότεροι κανίβαλοι από ιεραπόστολους σε κάποια όχθη, προστίθεται μια ποινή **(penalty)** στο κόστος για να αποτρέψει καταστάσεις αυτές.

Τελικά προκύπτει **2 * trips + 1 + penalty** όταν η βάρκα βρίσκεται ήδη στα αριστερά καθώς χρειάζεται +1 διαδρομή για να φτάσει στην δεξιά όχθη.
Ή **2 * trips + penalty** όταν η βάρκα βρίσκεται στην δεξιά όχθη.

3. Έλεγχος εγκυρότητας κατάστασης is_valid(self).

Ελέγχει αν ο αριθμός των ιεραποστόλων ή κανιβάλων στην αριστερή όχθη είναι αρνητικός. Υπολογίζει τον αριθμό των ανθρώπων στη δεξιά όχθη ως τη διαφορά από το συνολικό πλήθος **(N)**. Βεβαιώνει ότι ο αριθμός των ατόμων σε κάθε όχθη (αριστερή ή δεξιά) δεν ξεπερνά το συνολικό πλήθος. Ελέγχει αν σε οποιαδήποτε όχθη ο αριθμός των κανίβαλων δεν ξεπερνάει τον αριθμό των ιεραποστόλων.

4. Έλεγχος τελικής κατάστασης is_final(self).

Ελέγχει αν η κατάσταση είναι τελική. Για να θεωρηθεί μια κατάσταση τελική πρέπει στην αριστερή όχθη να μην υπάρχει κανένα άτομο και η βάρκα να βρίσκεται στην δεξιά όχθη.

5. Δημιουργία έπομενων καταστάσεων get_children(self).

Παράγει όλες τις δυνατές έγκυρες καταστάσεις (παιδιά) που μπορούν να προκύψουν από την τρέχουσα κατάσταση, εξετάζοντας όλες τις πιθανές μεταφορές ιεραποστόλων και κανιβάλων με τη βάρκα. Για κάθε συνδυασμό ατόμων (ιεραποστόλων και κανιβάλων) που μπορούν να χωρέσουν στη βάρκα και είναι έγκυρος (π.χ., οι ιεραπόστολοι να μην είναι λιγότεροι από τους κανίβαλους και να

υπάρχει τουλάχιστον ένα άτομο στη βάρκα), δημιουργεί μια νέα κατάσταση. Αν η βάρκα είναι στην αριστερή όχθη, τα άτομα αφαιρούνται από την αριστερή όχθη και προστίθενται στη δεξιά, και αντίστροφα. Κάθε νέα κατάσταση ελέγχεται για εγκυρότητα με τη μέθοδο **is_valid**, και μόνο οι έγκυρες καταστάσεις προστίθενται στη λίστα παιδιών, η οποία επιστρέφεται στο τέλος.

6. __hash__ Δημιουργήσει το hash ενός tuple που περιέχει τις τιμές (αριθμός ιεραποστόλων, αριθμός κανίβαλων, θέση της βάρκας).

7. __eq__ Καθορίζει πότε δύο αντικείμενα της κλάσης θεωρούνται ίσα.

8. __lt__ Καθορίζει ποια κατάσταση έχει μεγαλύτερη προτεραιότητα για ταξινόμηση.

9. print_state(self) Εμφανίζει τις θέσεις των ανθρώπων και στις βάρκας για κάθε κατάσταση.

A Star

Η μέθοδος χρησιμοποιεί δύο βασικές δομές δεδομένων το **open_dict**, που περιέχει καταστάσεις προς εξερεύνηση (με βάση το κόστος **f = g + h**), και το **closed_set**, που περιέχει τις ήδη εξερευνημένες καταστάσεις. Ο αλγόριθμος ξεκινά με την αρχική κατάσταση (**start_state**) στο **open_dict** και επαναλαμβάνεται μέχρι είτε να βρεθεί λύση (με την **is_final()**) είτε να εξαντληθούν οι καταστάσεις. Σε κάθε βήμα, επιλέγει την κατάσταση με το χαμηλότερο f από το **open_dict**, την αφαιρεί και την επεξεργάζεται. Οι επόμενες πιθανές καταστάσεις που επιστρέφονται από το **get_children()** ελέγχονται και στην περίπτωση που δεν έχουν εξερευνηθεί, προστίθενται στο **open_dict**, ενώ αν προκύπτει καλύτερο μονοπάτι για μια υπάρχουσα κατάσταση, το **f** ενημερώνεται. Ο περιορισμός στον αριθμό διασχίσεων εξασφαλίζεται με τον έλεγχο του **g** (πραγματικό κόστος). Επιστρέφει την τελική κατάσταση όταν επιτευχθεί, ή **None** αν δεν υπάρχει λύση εντός των περιορισμών.

Main

Αρχικά, ορίζεται το πλήθος των ιεραποστόλων και κανιβάλων (**N**), η χωρητικότητα της βάρκας (**M**), και ο μέγιστος αριθμός επιτρεπόμενων διασχίσεων (**K**). Στη συνέχεια, δημιουργείται η αρχική κατάσταση (initial_state) όπου όλοι οι ιεραπόστολοι και οι κανίβαλοι βρίσκονται στην αριστερή όχθη. Ο αλγόριθμος **A*** καλείται για να βρει μια λύση εντός του μέγιστου αριθμού διασχίσεων. Εάν βρεθεί λύση, η πορεία από την αρχική στην τελική κατάσταση ανακατασκευάζεται με χρήση του πατέρα (**_father**) κάθε κατάστασης και εκτυπώνεται βήμα-βήμα. Παράλληλα, εμφανίζεται ο συνολικός αριθμός των διασχίσεων και ο χρόνος που χρειάστηκε ο αλγόριθμος για να ολοκληρωθεί. Αν δεν βρεθεί λύση, ενημερώνεται ο χρήστης ότι ο περιορισμός στις διασχίσεις δεν επέτρεψε την επίλυση.

Παραδείγματα

- N = 3 M = 2 K = 100

Τελικό αποτέλεσμα.

```
"D:\Missionaries-and-Cannibals\Missionaries and Cannibals\.venv\Scripts\python.exe" "D:\Missionaries-
and-Cannibals\Missionaries and Cannibals\main.py"
Λύση βρέθηκε:
Αριστερή όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 3 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 0
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 1 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
-----
Βάρκα είναι στην δεξιά όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 1
Βάρκα έφτασε στην αριστερή όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 2 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 1
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 0 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 3
-----
Βάρκα είναι στην δεξιά όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 1
Βάρκα έφτασε στην αριστερή όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 1 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 2, Κανίβαλοι: 0
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 1, Κανίβαλοι: 1 || Δεξιά όχθη -> Ιεραπόστολοι: 2, Κανίβαλοι: 2
-----
Βάρκα είναι στην δεξιά όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 1, Κανίβαλοι: 1
Βάρκα έφτασε στην αριστερή όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 2, Κανίβαλοι: 2 || Δεξιά όχθη -> Ιεραπόστολοι: 1, Κανίβαλοι: 1
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 2, Κανίβαλοι: 0
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 2 || Δεξιά όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 1
-----
Βάρκα είναι στην δεξιά όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 1
Βάρκα έφτασε στην αριστερή όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 3 || Δεξιά όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 0
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 1 || Δεξιά όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 2
-----
Βάρκα είναι στην δεξιά όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 1
Βάρκα έφτασε στην αριστερή όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 2 || Δεξιά όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 1
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 0, Κανίβαλοι: 2
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 0 || Δεξιά όχθη -> Ιεραπόστολοι: 3, Κανίβαλοι: 3
-----
Συνολικές μετακινήσεις: 11
Συνολικός χρόνος ολοκλήρωσης: 0.0011 δευτερόλεπτα

Process finished with exit code 0
```

Διάσχιση καταστάσεων από την A*

```
g: 1  m: 0  c: 2 h: 8
g: 1  m: 1  c: 1 h: 8
g: 1  m: 0  c: 1 h: 10
g: 2  m: 0  c: 1 h: 11
g: 3  m: 0  c: 2 h: 6
g: 4  m: 0  c: 1 h: 9
g: 5  m: 2  c: 0 h: 4
g: 6  m: 1  c: 1 h: 9
g: 7  m: 2  c: 0 h: 4
g: 8  m: 0  c: 1 h: 7
g: 9  m: 0  c: 2 h: 2
g: 10  m: 0  c: 1 h: 1
g: 10  m: 1  c: 0 h: 1
g: 11  m: 0  c: 2 h: 0
```

g = το βάθος.

m = ο αριθμός των ιεραποστόλων στην βάρκα.

c = ο αριθμός των κανίβαλων στηνβάρκα.

h = το κόστος της ευρετικής.

Παρατηρούμε πως στην πρώτη διάσχιση και στην δέκατη, υπάρχουν 2 καταστάσεις που έχουν ίδια τιμή ευρετικής. Αυτό σημαίνει πως και οι δύο καταστάσεις είναι εξίσου σωστές.

- N>3 M=2 K=5

```
g: 1 m_l 4 c_l 2  m_b: 0  c_b: 2 h: 12
g: 1 m_l 3 c_l 3  m_b: 1  c_b: 1 h: 12
g: 1 m_l 4 c_l 3  m_b: 0  c_b: 1 h: 14
g: 2 m_r 0 c_r 1 m_b: 0  c_b: 1 h: 15
g: 3 m_l 4 c_l 1  m_b: 0  c_b: 2 h: 10
g: 4 m_r 0 c_r 2 m_b: 0  c_b: 1 h: 13
g: 5 m_l 4 c_l 0  m_b: 0  c_b: 2 h: 8
g: 5 m_l 2 c_l 2  m_b: 2  c_b: 0 h: 8
g: 6 m_r 0 c_r 3 m_b: 0  c_b: 1 h: 11
g: 6 m_r 1 c_r 1 m_b: 1  c_b: 1 h: 13
Δεν βρέθηκε λύση εντός των επιτρεπόμενων διασχίσεων.
Συνολικός χρόνος ολοκλήρωσης: 0.0000 δευτερόλεπτα
```

Σε αυτή την περίπτωση παρατηρούμε πως δεν υπάρχει λύση καθώς μετά την πέμπτη διάσχιση δεν μπορεί να υπάρξει αποδεκτή κατάσταση.

- N = 2 M = 4 K=5

```
g: 1 m_l 0 c_l 0  m_b: 2  c_b: 2 h: 0
Λύση βρέθηκε:
Αριστερή όχθη -> Ιεραπόστολοι: 2, Κανίβαλοι: 2 || Δεξιά όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 0
-----
Βάρκα είναι στην αριστερή όχθη και οι επιβάτες είναι -> Ιεραπόστολοι: 2, Κανίβαλοι: 2
Βάρκα έφτασε στην δεξιά όχθη.
Αριστερή όχθη -> Ιεραπόστολοι: 0, Κανίβαλοι: 0 || Δεξιά όχθη -> Ιεραπόστολοι: 2, Κανίβαλοι: 2
-----
Συνολικές μετακινήσεις: 1
Συνολικός χρόνος ολοκλήρωσης: 0.0000 δευτερόλεπτα
```

Συμπέρασμα

Η υλοποίηση του γρίφου των Ιεραποστόλων και Κανιβάλων μέσω του αλγορίθμου **A*** αποδεικνύει την αποτελεσματικότητα της συνδυασμένης χρήσης πραγματικού κόστους και ευρετικής συνάρτησης στην επίλυση προβλημάτων περιορισμών. Η χρήση κατάλληλων δομών δεδομένων και η εφαρμογή σαφών περιορισμών διασφαλίζουν την ακρίβεια και την αποδοτικότητα της λύσης, ακόμα και για σύνθετες παραλλαγές του προβλήματος. Το μοντέλο αυτό μπορεί να επεκταθεί και να προσαρμοστεί σε άλλες εφαρμογές, παρέχοντας χρήσιμες γνώσεις για την επίλυση παρόμοιων προβλημάτων στον τομέα της τεχνητής νοημοσύνης και των αλγορίθμων αναζήτησης.