

Τεχνητή Νοημοσύνη

1^η Προγραμματιστική εργασία

Ματσούκας Ιωάννης-3170106

Πολυχρονάκης Ανδρέας-3170140

Τσελές Άγγελος-3170160

Ακαδημαϊκό έτος:2020-2021

Το πρόβλημα που επιλέξαμε να ασχοληθούμε είναι η διάσχιση της γέφυρας(BridgeCrossing puzzle)

Ο αλγόριθμος που υλοποιήσαμε είναι ο A*.Ξεκινάμε από μια κατάσταση,ουσιαστικά την αρχική μας κατάσταση και επεκτείνουμε τον κόμβο με το μικρότερο $f(n)=h(n)+g(n)$

Όπου $h(n)$:η ευρετική μας συνάρτηση

Σχετικά με την ευρετική:Δεν έχουν όλες οι καταστάσεις την ίδια αξία.Αυτές οι οποίες έχουν φέρει τα άτομα με τους μεγαλύτερους χρόνους στην άλλη όχθη είναι και πιο κοντά στην λύση με την λογική κυριώς ότι έχουν και μεγαλύτερο βάρος δηλαδή έχουν αρκετα μεγάλο κόστος μεταφοράς αλλά και τα άτομα με το μικρότερο χρόνο πηγαίνουν από την μια όχθη στην άλλη οπότε είναι προσωρινοί.

$g(n)$:Το κόστος μετάβασης από τον έναν κόμβο στον επόμενο.Ουσιαστικά σαν κόστος έχουμε πάρει το χρόνο που χρειάζεται ένα μελος να πάει από την μια οχθη στην άλλη η δυο μέλη αντίστοιχα(χρόνος του πιο αργού)

Για δική μας διευκόλυνση προσπαθήσαμε να φέρουμε το πρόβλημα όσο πιο κοντά σε μια ρεαλιστική προσέγγιση δηλαδή:

Φτιάξαμε μια κλάση State.Το βασικό αυτής της κλάσης είναι ότι κρατάει έναν δισδιάστατο πίνακα,οπου στην αρχική κατάσταση(Initial State) όλα τα μέλη της οικογένειας(Κάθε κελί του πίνακα κρατάει ένα αντικείμενο τύπου Person) είναι στην μια πλευρά ακριβώς όπως και την γέφυρα όλα τα μέλη είναι από την μια όχθη και πρέπει να περάσουν στην απέναντι.

Για να γίνει αυτό υλοποιούμε τις εξής μεθόδους

`canCrossOnePerson()`->ελέγχει με βάση ποιος έχει την λαμπα στην κάθε όχθη αν είναι στην initial όχθη τότε θα καλέσει την `crossTwoPeople()`

canCrossTwoPeople()->ελέγχει με βάση ποιος έχει την λαμπα στην κάθε όχθη αν είναι στην final όχθη τότε θα καλέσει την crossOnePerson()

crossTwoPeople()->επιστρέφει ένα arraylist με όλες τις δυνατές καταστάσεις-παιδιά από μια κατάσταση υλοποιώντας την swapTwo

crossOnePerson()->επιστρέφει ένα arraylist με όλες τις δυνατές καταστάσεις-παιδιά από μια κατάσταση υλοποιώντας την swapOne

swapOne()->Καλείται από την μέθοδο crossOnePerson() και μετακινεί ένα άτομο από την μια όχθη στην άλλη

swapTwo()->Καλείται από την μέθοδο crossTwoPeople() και μετακινεί δύο άτομα από την μια όχθη στην άλλη

Αυτές οι μέθοδοι υλοποιούνται από την getChildren() η οποία φτιάχνει ουσιαστικά τα παιδιά της κάθε κατάστασης με την βοήθεια των παραπάνω μεθόδων

Στην κλάση spaceSearcher υλοποιείται ο αλγόριθμος A* με ανοικτό σύνολο

Ξεκινάμε βάζοντας στο open set την initial state.Όσο το open set δεν είναι άδειο(δηλαδή έχουμε καταστάσεις να επεκτείνουμε),τις ταξινομούμε με βάση το f και επιλέγουμε την κατάσταση με το μικρότερο f.Την αφαιρούμε από το σύνολο και την επεκτείνουμε.Δημιουργούμε όλα τα παιδιά της,ελέγχουμε αν υπάρχουν στο open set.Αν δεν υπάρχουν τις προσθέτουμε αλλιώς αν υπάρχει και έχει και μικρότερη f τότε απλά δεν ασχολούμαστε με αυτήν την κατάσταση

Το κύριο πρόβλημα(bridgeCrossing Puzzle με 5 άτομα με τους fixed χρόνους)

```
0
-----
1 3 6 8 12
0 0 0 0 0
-----

3
-----
0 0 6 8 12
1 3 0 0 0
-----

4
-----
1 0 6 8 12
0 3 0 0 0
-----

16
-----
1 0 6 0 0
0 3 0 8 12
-----

19
-----
1 3 6 0 0
0 0 0 8 12
-----

22
-----
0 0 6 0 0
1 3 0 8 12
-----

23
-----
1 0 6 0 0
0 3 0 8 12
-----
```

```
29
-----
0 0 0 0 0
1 3 6 8 12
-----

To lynei se xrono: 29
A* with closed set search time: 0.003 sec.

Process finished with exit code 0
```

Επίσης, να επισημάνουμε ότι δεν υπάρχει πάντα λύση στο πρόβλημα δηλαδή να βρει λύση με χρόνο μικρότερο ή ίσο από το άθροισμα των χρόνων των ατόμων όπως στην περίπτωση πάνω όπου το άθροισμα είναι 30

Μια τέτοια περίπτωση που παρατηρήσαμε είναι αυτή με 4 άτομα και χρόνους 1,2,3,4

```
Πληκτρολόγησε τον αριθμο των χαρακτηρων: 4
Πληκτρολόγησε τον χρόνο του χαρακτήρα 1 : 1
Πληκτρολόγησε τον χρόνο του χαρακτήρα 2 : 2
Πληκτρολόγησε τον χρόνο του χαρακτήρα 3 : 3
Πληκτρολόγησε τον χρόνο του χαρακτήρα 4 : 4
-----
1 2 3 4
0 0 0 0
-----

Could not find solution
A* with open set search time: 0.0 sec.
```

Καθώς μηχανικά βρήκαμε ότι η βέλτιστη λύση είναι σε χρόνο 11

Τέλος ένα ακόμα παράδειγμα με τυχαία νούμερα και διαφορετική τιμή του N

```
0
-----
1 3 10 15 18 25 45 80 100 203
0 0 0 0 0 0 0 0 0 0
-----

354
-----
0 0 0 0 0 0 0 0 0 0
1 3 10 15 18 25 45 80 100 203
-----
```