

# **Artificial Intelligence Project**

## **Μέλη ομάδας:**

1. Αικατερίνη Τσιτσιμίκλη AM 4821
2. Σολδάτου Χριστίνα Ολυμπία AM 4001
3. Πανταζής Δημοσθένης AM 4136

## 1.Εργαλεία

- Java και C

## 2.Γενικά

Για την μοντελοποίηση του προβλήματος μας ακολουθήσαμε πιστά την γενική δομή του αλγόριθμου αναζήτησης χρησιμοποιώντας σαν γνώμονα το υλικό του μαθήματος. Αρχικά τοποθετούμε την αρχική κατάσταση στο μέτωπο αναζήτησης. Εάν το μέτωπο αναζήτησης είναι άδειο τότε ο αλγόριθμος σταματάει. Στην συνέχεια παίρνουμε την πρώτη σε σειρά κατάσταση του μετώπου. Ένα προαιρετικό βήμα είναι η μετάβαση στο βήμα 2 (επόμενο βήμα) αν και μόνο αν η τελευταία κατάσταση είναι μέρος του κλειστού συνόλου. Αντίθετα, αν η κατάσταση είναι τελική τότε τυπώνεται η λύση και ο αλγόριθμος τερματίζει. Στην συνέχεια εφαρμόζουμε τους τελεστές μετάβασης για την παραγωγή των καταστάσεων -παιδιά (γείτονες). Τοποθετούμε τις νέες καταστάσεις-παιδιά στο μέτωπο αναζήτησης και προαιρετικά βάζουμε και την κατάσταση γονέα σε ένα κλειστό σύνολο. Καταληκτικά φτάνουμε στο βήμα 2ο και αρχίζει ξανά η διαδικασία από την αρχή.

## 3. Αναζήτηση Ομοιόμορφου Κόστους (UCS)

Η μοντελοποίηση γίνεται στην κλάση AI\_1i η οποία αναπαριστά οποιοδήποτε πρόβλημα μεγέθους N (Το N πρέπει να δηλωθεί στην κλάση AI\_1i. Η αρχική τιμή του είναι N = 6).

Για την λύση του προβλήματος, ο χρήστης καλείται να δώσει μερικές συντεταγμένες των κελιών S, G1 και G2. Όπου S το αρχικό κελί και όπου G1 και G2 τα τελικά. Τα κελιά αυτά καθίστανται κατειλημμένα ώστε να μπορεί να δημιουργηθεί ο λαβύρινθος στο οποίο θα κινηθεί το ρομπότ. Στην συνέχεια τυπώνεται ο πίνακας που αναπαριστά τον λαβύρινθο, τις επεκτάσεις, το συνολικό κόστος της αναζήτησης καθώς και η βέλτιστη διαδρομή. Δίπλα παραθέτουμε ένα παράδειγμα με συντεταγμένες:

S=(1,4)

G1 = (1,5)

G2 =(2,4)

```
give coordinates of S:
1
4
give coordinates of G1:
1
5
give coordinates of G2:
2
4
- / 1   i   - / 1   - / 1   - / 2   - / 2
- / 3   - / 4   - / 2   - / 3   S / 2   A / 1
- / 1   - / 2   - / 3   - / 3   B / 3   - / 4
   i   - / 4   - / 2   - / 4   - / 2   - / 4
- / 3   - / 1   - / 2   - / 1   - / 1   - / 4
- / 3   - / 4   - / 1   - / 4   - / 4   - / 4
total cost = 2.000000
optimal root: (1,4) -> (2,4)
externals: 0

Would you like to continue? (yes or no)
```

Στον αλγόριθμο UCS πραγματοποιείται η σχέση  $\min[ g(n) ]$ . Όπου  $g(n)$  το κόστος μετάβασης από το κελί S μέχρι την n. Για να βρεθεί η διαδρομή με το ελάχιστο κόστος μεταβαίνουμε σε γειτονικές καταστάσεις. Οι θέσεις αυτές εντός του λαβυρίνθου θα είναι κατακόρυφες, οριζόντιες και διαγώνιες εφόσον δεν υπάρχουν εμπόδια.