Τεχνητή Νοημοσύνη
Εαρινό Εξάμηνο 2017
Διδάσκων: Α. Λύκας

Εργαστηριακή Ασκηση 1 (αλγόριθμος Α*)

Να κατασκευάσετε πρόγραμμα αναζήτησης για την επίλυση του ακόλουθου προβλήματος **πλοήγησης ρομπότ σε λαβύρινθο**:

Κατασκευή λαβύρινθου: Θεωρούμε ένα πλέγμα με NxN θέσεις, κάποιες από τις οποίες είναι ελεύθερες, ενώ οι υπόλοιπες περιέχουν εμπόδια και δεν μπορούμε να τις επισκεφθούμε. Κάθε θέση (x,y) χαρακτηρίζεται ως ελεύθερη ή όχι αποφασίζοντας ανεξάρτητα με πιθανότητα p. Τα N και p καθορίζονται στην αρχή του προγράμματος.

Θέλουμε να ορίσουμε τη **διαδρομή** (εάν υπάρχει) που πρέπει να ακολουθήσει το ρομπότ ξεκινώντας από μια αρχική κατάσταση (S) ώστε **να περάσει διαδοχικά** από δύο συγκεκριμένες καταστάσεις στόχους G1 και G2. Η σειρά με την οποία θα επισκεφθεί τις G1 και G2 εξαρτάται από την αρχική κατάσταση S. Οι συντεταγμένες (x,y) των S, G1 και G2 δίνονται από τον χρήστη στην αρχή του προγράμματος. Το ρομπότ μπορεί κάθε φορά να μετακινείται **είτε οριζόντια είτε κατακόρυφα** σε μία **γειτονική** ελεύθερη θέση. **Κάθε οριζόντια κίνηση έχει κόστος 0.5 και κάθε κατακόρυφη έχει κόστος 1.**

Το πρόβλημα διαιρείται σε δύο υποπροβλήματα: Το πρώτο υποπρόβλημα αναζήτησης αφορά στη βέλτιστη μετακίνηση του ρομπότ από την αρχική κατάσταση S στην πλησιέστερη (με βάση την συνάρτηση κόστους) από τις καταστάσεις στόχους (G1 ή G2). Στη συνέχεια, το δεύτερο υποπρόβλημα αναζήτησης αφορά στη βέλτιστη μετακίνηση του ρομπότ από την κατάσταση στόχο την οποία βρίσκει το πρώτο υποπρόβλημα (G1 ή G2) στην άλλη κατάσταση στόχο (G2 ή G1 αντίστοιχα).

Να υλοποιήσετε αναζήτηση A^* για κάθε υποπρόβλημα χρησιμοποιώντας όσο το δυνατόν καλύτερη **αποδεκτή ευρετική συνάρτηση** h(n) για να καθορίσετε τη συνολική διαδρομή του ρομπότ Να εξετάσετε διάφορες τιμές του N και του p. Στο τέλος θέλουμε να τυπώνεται ο λαβύρινθος, οι καταστάσεις S, G1 και G2, το συνολικό μονοπάτι που βρήκατε, το κόστος του συνολικού μονοπατιού αυτού, καθώς και ο αριθμός των επεκτάσεων που έγιναν (και στις δύο αναζητήσεις).

Εργαστηριακή Ασκηση 2 (ικανοποίηση περιορισμών με Προσομοιούμενη Ανόπτηση)

Να υλοποιήσετε τη μέθοδο της Προσομοιούμενης Ανόπτησης (SA) και να τη χρησιμοποιήσετε για την κατασκευή προγράμματος επίλυσης **puzzle Hitori** (https://en.wikipedia.org/wiki/Hitori).

4	8	1	6	3	2	5	7
3	6	7	2	1	6	5	4
2	3	4	8	2	8	6	1
4	1	6	5	7	7	3	5
7	2	3	1	8	5	1	2
3	5	6	7	3	1	8	4
6	4	2	3	5	4	7	8
8	7	1	4	2	3	5	6

Αρχικά δεδομένα

■	8	■	6	3	2	■	7
3	6	7	2	1	■	5	4
■	3	4	■	2	8	6	1
4	1	■	5	7	■	3	■
7	■	3	■	8	5	1	2
■	5	6	7	■	1	8	■
6	■	2	3	5	4	7	8
8	7	1	4	■	3	■	6

Λύση

Στο πρόβλημα αυτό δίνεται ως είσοδος ένας πίνακας MxM (π.χ. M=9) από τετράγωνα που περιέχουν ακεραίους από 1 έως K (K<=M).

Στόχος είναι να σβηστούν οι αριθμοί από κάποια τετράγωνα (δηλ. να 'μαυρίσουν' κάποια τετράγωνα) ώστε να ικανοποιούνται οι παρακάτω περιορισμοί:

1) Να μην εμφανίζεται κάποιος αριθμός περισσότερες από μία φορές στην ίδια γραμμή ή στην ίδια στήλη.
2) Να μην υπάρχουν γειτονικά μαύρα τετράγωνα (οριζόντια ή κάθετα).
3) Τα ‘λευκά’ τετράγωνα (δηλ. τα τετράγωνα με αριθμό) να είναι πλήρως συνδεδεμένα: για κάθε λευκό τετράγωνο να υπάρχει μονοπάτι (μετακινούμενοι οριζόντια ή κάθετα σε γειτονικά λευκά τετράγωνα) προς οποιοδήποτε άλλο λευκό τετράγωνο.

Το πρόγραμμα θα παίρνει ως είσοδο τον αρχικό πίνακα με τους αριθμούς (βλ. παραπάνω σχήμα) και θα αναζητά την τελική λύση επιλύοντας με τη μέθοδο SA το πρόβλημα ικανοποίησης περιορισμών που προκύπτει: θεωρούμε ότι **κάθε τετράγωνο είναι μια μεταβλητή που έχει την τιμή ‘λευκό’ ή ‘μαύρο’**. Στόχος της αναζήτησης είναι η ανάθεση επιτρεπτών τιμών σε όλες τις μεταβλητές ώστε να μην παραβιάζεται κάποιος περιορισμός. Να ορίσετε κατάλληλη συνάρτηση κόστους ανάλογη με το πλήθος των περιορισμών που παραβιάζονται σε μία κατάσταση.

Να φτιαχτούν δύο προγράμματα:

A) Το πρόγραμμα CSP1 που θα λύνει το puzzle χωρίς τον περιορισμό 3.
B) Το πρόγραμμα CSP2 που θα λύνει το puzzle με τον περιορισμό 3.

Οι παράμετροι του αλγορίθμου SA να καθοριστούν ως εξής: i) αρχική θερμοκρασία T_0 = 10.0, ii) αριθμός δοκιμών ανά θερμοκρασία 2N (όπου N ο αριθμός των μεταβλητών), iii) ρυθμός μείωσης της θερμοκρασίας: $T_{k+1}=0.999\ T_k$, iv) τερματισμός μόλις φθάσουμε σε λύση ή εάν έχουν γίνει 6N διαδοχικές ανεπιτυχείς δοκιμές μετάβασης. Αρχικά όλες οι μεταβλητές να είναι ‘λευκές’, είτε εναλλακτικά η αρχική κατάσταση να καθορίζεται τυχαία. Θα πρέπει να τυπώνεται η τελική κατάσταση και το κόστος της.