Τεχνητή Νοημοσύνη

1115201800006_project1.pdf

Αναγνωστόπουλος Αθανάσιος

AM: 1115201800006

Πρόβλημα 2

Ο συνολικός αριθμός των κόμβων που επεκτείνονται με τον αλγόριθμο πρώτα σε βάθος με επαναληπτική εκβάθυνση (IDS ή IDDFS) είναι:

$(d)b + (d-1)b^2 + ... + 3b^{d-2} + 2b^{d-1} + b^d$

Άρα ο μεγαλύτερος αριθμός κόμβων που μπορούν να δημιουργηθούν είναι ο τύπος στην χειρότερη περίπτωση.

Δηλαδή είναι η σειρά: $\sum_{i=1}^{d}(d+1-i)b^i$

Αντίθετα ο μικρότερος αριθμός κόμβων που μπορούν να δημιουργηθούν είναι για d = g, δηλαδή όταν:

$(g)b + (g-1)b^2 + ... + 3b^{g-2} + 2b^{g-1} + b^g$

Δηλαδή είναι η σειρά: $\sum_{i=1}^{g}(g+1-i)b^i$

Πρόβλημα 3

(α)

Η συνάρτηση δεν είναι παραδεκτή.

Αντιπαράδειγμα: Για παράδειγμα στον κόμβο b3 το πραγματικό κόστος εύρεσης από τον o103 είναι 4 αλλά σύμφωνα με την ευρετική έχουμε h(b3) = 17. Επομένως αφού ισχύει h(b3) > πραγματικό κόστος η ευρετική υπερεκτιμάει το κόστος και τελικά η συνάρτηση δεν είναι παραδεκτή.

Απαραίτητη προϋπόθεση για να είναι μια συνάρτηση συνεπής είναι να είναι παραδεκτή. Εφόσον δεν είναι παραδεκτή ⇨ δεν είναι ούτε συνεπής.

(β)

- BFS: o103 ⇨ b3 ⇨ o109 ⇨ ts ⇨ b1 ⇨ b4 ⇨ o111 ⇨ o119 ⇨ mail ⇨ b2 ⇨ c2 ⇨ o123 ⇨ storage ⇨ c1 ⇨ c3 ⇨ o125 ⇨ r123
- DFS: o103 ⇨ ts ⇨ mail ⇨ o109 ⇨ o119 ⇨ storage ⇨ o123 ⇨ r123

- Αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση:

  $1^{η}$:
  o103

  $2^{η}$:
  o103 ⇨ ts ⇨ o109 ⇨ b3

  $3^{η}$:
  o103 ⇨ ts ⇨ mail ⇨ o109 ⇨ o119 ⇨ o111 ⇨ b3 ⇨ b4 ⇨ b1

$4^{η}$:
o103 ⇨ ts ⇨ mail ⇨ o109 ⇨ o119 ⇨ storage ⇨ o123 ⇨ o111 ⇨ b3 ⇨ b4 ⇨ b1 ⇨ c2 ⇨ b2

$5^{η}$:
o103 ⇨ ts ⇨ mail ⇨ o109 ⇨ o119 ⇨ storage ⇨ o123 ⇨ r123

- Άπληστη αναζήτηση πρώτα στον καλύτερο με ευρετική συνάρτηση h:

  o123 ⇨ b3 ⇨ b1 ⇨ c2 ⇨ c1 ⇨ c3 ⇨ b2 ⇨ b4 ⇨ ts ⇨ o109 ⇨ o119 ⇨ o123 ⇨ r123

- A* με ευρετική συνάρτηση h:

  o123 ⇨ b3 ⇨ b1 ⇨ c2 ⇨ c1 ⇨ b2 ⇨ b4 ⇨ c3 ⇨ts ⇨ o109 ⇨ o119 ⇨ mail ⇨ o123 ⇨ r123

Πρόβλημα 4

(α)

Έστω ότι το ρομπότ έχει να μεταφέρει n πακέτα

Ένα πρόβλημα αναζήτης ορίζεται από:

1. Αρχική κατάσταση StartState η οποία θα αποτελείται από το δωμάτιο που βρίσκεται αρχικά το ρομπότ, δηλαδή το mail, και ακόμα θα περιέχει λίστα από tuples π.χ. (roomx,destinationx) όπου roomx το δωμάτιο όπου βρίσκεται το πακέτο x και destinationx ο προορισμός του πακέτου x.

2. GoalState ο στόχος θα είναι όταν το ρομπότ έχει επιστρέψει στο δωμάτιο mail και ο αριθμός των πακέτων που έχουν απομείνει είναι 0 (NumOfPacks == 0).

3. A set of states, όπου θα αποτελείται από ένα state και το κόστος του π.χ. (statex , costx) και statex = (roomx,destinationx).

4. Successor Function η οποία θα παίρνει ως όρισμα ένα state και θα επιστρέφει τις επόμενες διαδρομές που μπορεί να κάνει το ρομπότ στη συνέχεια.

(β)

Μια ευρετική συνάρτηση θα ήταν ο υπολογισμός του κόστους της απόστασης από την θέση του ρομπότ μέχρι την θέση του πακέτου + το κόστος της απόστασης από την θέση του πακέτου μέχρι τον προορισμό του.

Η ευρετική είναι παραδεκτή εφόσον δεν υπερεκτιμά το συνολικό κόστος.

Πρόβλημα 5

	Πλήρης	Βέλτιστος
(α)	Όχι	Όχι
(β)	Όχι	Όχι
(γ)	Όχι	Όχι
(δ)	Ναι υπο την συνθήκη ότι όλα τα κόστοι είναι θετικά	Ναι εάν όλα τα κόστοι είναι θετικά και ίδια

Ο έλεγχος ότι οι δύο αναζητήσεις συναντιούνται σε κάθε μία από τις περιπτώσεις (α) – (δ) μπορεί να γίνει ελέγχοντας τα frontier και των δύο αναζητήσεων. Θα ελέγξουμε δηλαδή αν περιέχουν κοινά στοιχεία και θα επιλέξουμε όποιο από τα δύο έχει μικρότερο κόστος.