Αλγόριθμοι Ευρετικής Αναζήτησης

(heuristic search)

Ευρετικός μηχανισμός (heuristic) είναι μία στρατηγική, βασισμένη στη γνώση για το συγκεκριμένο πρόβλημα, η οποία χρησιμοποιείται ως βοήθημα στην ταχύτερη επίλυσή του.

- Ο ευρετικός μηχανισμός υλοποιείται με **ευρετική συνάρτηση** (heuristic function), που έχει πεδίο ορισμού το σύνολο των καταστάσεων ενός προβλήματος και η τιμή της ευρετικής συνάρτησης h(n) εκφράζει το πόσο απέχει μία κατάσταση n στην πλησιέστερη τελική.
- Η ευρετική τιμή δεν είναι η πραγματική τιμή της (συνήθως άγνωστης) απόστασης (a(n)) από την πλησιέστερη τερματική κατάσταση, αλλά μία **εκτίμηση (estimate*)*** που πολλές φορές μπορεί να μην είναι ακριβής.
- Την ευρετική συνάρτηση την επινοεί ο χρήστης. Μπορούν να επινοηθούν πολλές ευρετικές συναρτήσεις για ένα πρόβλημα.
- Μια ευρετική συνάρτηση h(n) πρέπει να υποχρεωτικά να είναι μεγαλύτερη του μηδενός για μη τελικές καταστάσεις και ίση με το μηδέν για τις τελικές καταστάσεις.

Απληστη Αναζήτηση (Greedy Search)

Επεκτείνεται ο κόμβος του μετώπου αναζήτησης με το μικρότερο h(n)

Ο αλγόριθμος GS

1. Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Πάρε την πρώτη σε σειρά κατάσταση από το μέτωπο αναζήτησης.
4. Αν η κατάσταση είναι μέλος του κλειστού συνόλου τότε πήγαινε στο 2. (προαιρετικό)
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση και σταμάτα.
6. Εφάρμοσε τους τελεστές μεταφοράς για να παράγεις τις καταστάσεις-παιδιά.
7. Υπολόγισε την ευρετική συνάρτηση σε κάθε παιδί.
8. Βάλε τις καταστάσεις-παιδιά στο μέτωπο αναζήτησης.
9. Αναδιάταξε το μέτωπο αναζήτησης, έτσι ώστε η κατάσταση με την καλύτερη ευρετική τιμή να είναι πρώτη.
10. Βάλε τη κατάσταση-γονέα στο κλειστό σύνολο. (προαιρετικό)
11. Πήγαινε στο βήμα 2.

Ο αλγόριθμος Απληστης Αναζήτησης
Σχόλια

- Προσπαθεί να δώσει μια γρήγορη λύση σε κάποιο πρόβλημα. Το αν τα καταφέρει ή όχι εξαρτάται από την ευρετική συνάρτηση.
- Είναι πλήρης, αλλά όχι βέλτιστη

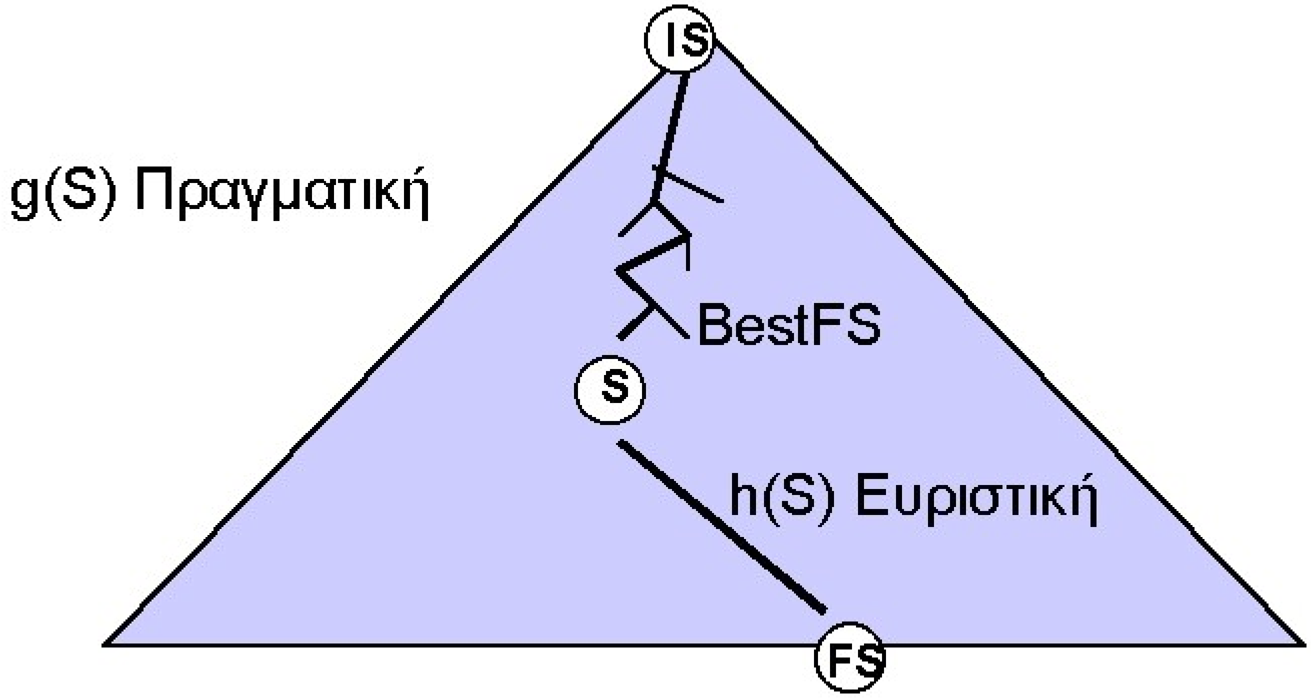
Ο Αλγόριθμος Άλφα-Άστρο (Α*)

Στον *αλγόριθμο Α (*Άλφα Άστρο*) επεκτείνεται η κατάσταση του Μετώπου Αναζήτησης με το μικρότερο:**

$$e(n) = g(n) + h(n)$$

η g(n) δίνει την απόσταση της n από την αρχική κατάσταση, η οποία είναι πραγματική και γνωστή

η h(n) δίνει την εκτίμηση της απόστασης της n από την τελική κατάσταση μέσω μιας ευρετικής συνάρτησης



- Αν για μια ευρετική συνάρτηση h(n) ισχύει ότι για κάθε κατάσταση n η τιμή h(n) είναι μικρότερη ή το πολύ ίση με την πραγματική απόσταση a(n) της n από την πλησιέστερη τελική κατάσταση, τότε η h(n) ονομάζεται **αποδεκτή (admissible).**
- **Αν η h(n) είναι αποδεκτή (h(n)<=a(n) για κάθε n), τότε η μέθοδος A* είναι πλήρης και βέλτιστη.**

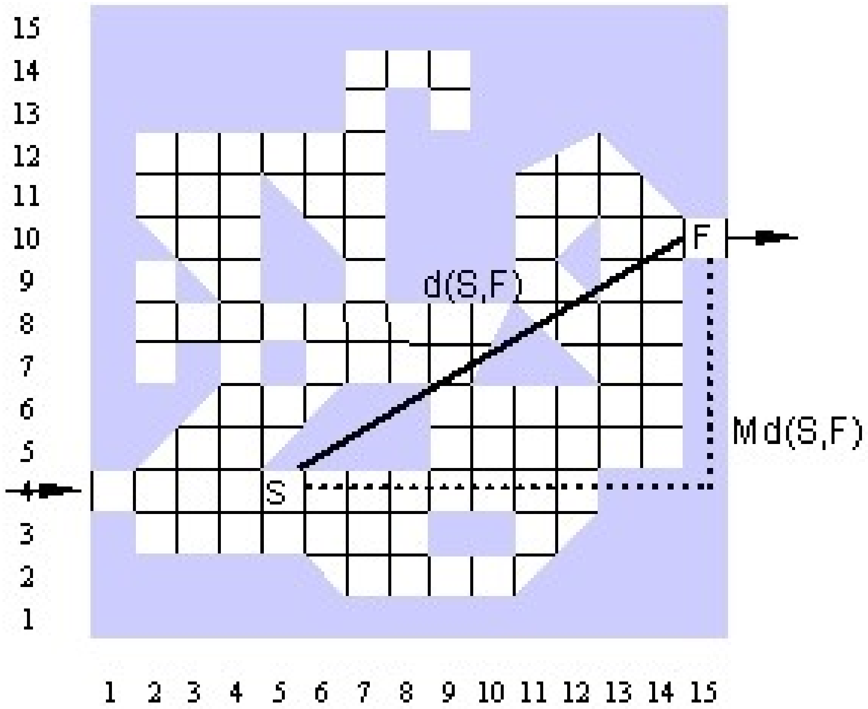
Ευρετικές συναρτήσεις σε λαβύρινθο

- Ευκλείδειος απόσταση (Euclidian distance):

$$d(S,F) = \sqrt{(X_S - X_F)^2 + (Y_S - Y_F)^2}$$

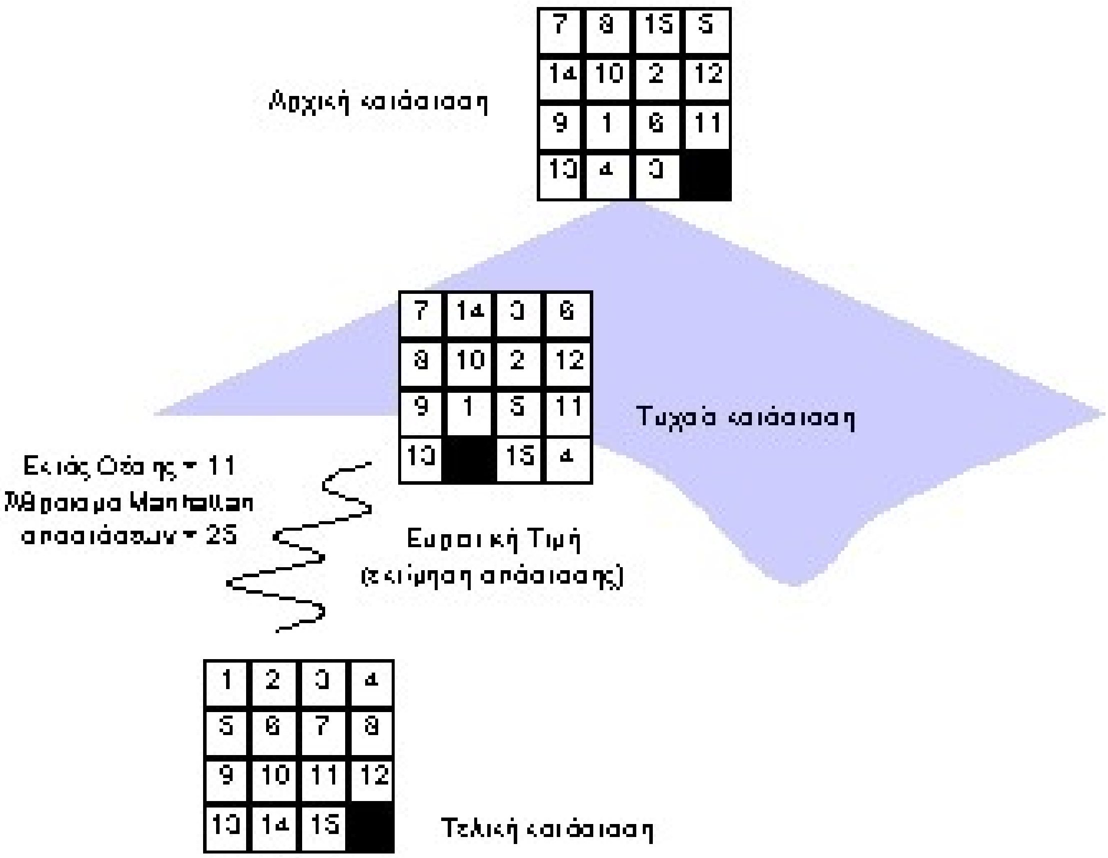
- Απόσταση Manhattan (Manhattan distance):

$$Md(S,F) = |X_S - X_F| + |Y_S - Y_F|$$



Ευρετικός μηχανισμός και συναρτήσεις στο N-Puzzle

- Πόσα πλακίδια βρίσκονται εκτός θέσης.
- Το άθροισμα των αποστάσεων Manhattan κάθε πλακιδίου από την τελική του θέση.

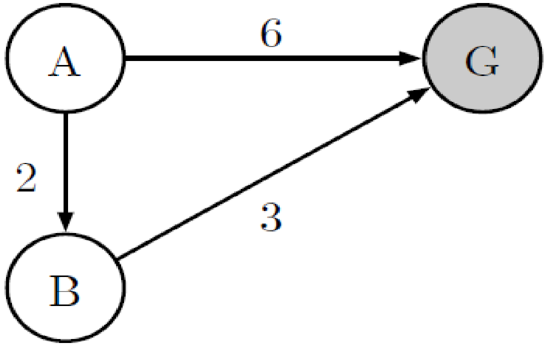


Στρατηγική Κατασκευής Αποδεκτών Ευρετικών Συναρτήσεων

- Αφαιρώντας περιορισμούς από το πρόβλημά μας κατασκευάζουμε ένα '**χαλαρωμένο**' (relaxed) πρόβλημα για το οποίο μπορούμε να υπολογίσουμε ακριβώς την συνάρτηση a(n) της απόστασης κάθε κατάστασης n από την πλησιέστερη Τ.Κ.
- **Η συνάρτηση a(n) του χαλαρωμένου προβλήματος αποτελεί αποδεκτή ευρετική συνάρτηση h(n) για το αρχικό πρόβλημα.**

- Οσο μεγαλύτερες τιμές έχει μια **αποδεκτή** ευρετική συνάρτηση, τόσο καλύτερη θεωρείται.
- Αν $h_1(n)$, $h_2(n)$, …, $h_K(n)$ είναι **αποδεκτές** ευρετικές συναρτήσεις για ένα πρόβλημα, τότε και η $h(n)=\max(h_1(n), h_2(n), \ldots, h_K(n))$ είναι **αποδεκτή** ευρετική συνάρτηση.

Ασκηση 1



	h(A)	h(B)	h(G)
I	4	1	0
II	5	4	0
III	6	3	0
IV	5	2	0

Θεωρούμε το γράφημα μεταβάσεων που φαίνεται στο διπλανό σχήμα (G είναι η τελική κατάσταση). Σε κάθε ακμή αναγράφεται το κόστος μετάβασης. Ορίζουμε τέσσερις ευρετικές συναρτήσεις, καθεμιά από τις οποίες αντιστοιχεί σε μια γραμμή του πίνακα (στην πρώτη γραμμή η $h_1(n)$, κλπ).

(α1) Είναι οι συναρτήσεις αυτές αποδεκτές;

(α2) Υπάρχει κάποια αποδεκτή συνάρτηση που είναι καλύτερη από τις υπόλοιπες;

Λύση

(1) Έστω $a(n)$ το ελάχιστο κόστος μονοπατιού από μια κατάσταση n έως την τελική κατάσταση G. Παρατηρούμε ότι: $a(A)=5$ (=2+3), $a(B)=3$, $a(G)=0$.

Για να είναι η $h(n)$ αποδεκτή θα πρέπει $h(n) \leq a(n)$ για κάθε κατάσταση n.

Οπότε έχουμε ότι:

- $h_1(n)$ αποδεκτή
- $h_2(n)$ **δεν είναι** αποδεκτή (διότι $h_2(B)>a(B)$)
- $h_3(n)$ **δεν είναι** αποδεκτή (διότι $h_3(A)>a(A)$)
- $h_4(n)$ παραδεκτή.

(2) Από τις παραπάνω δύο αποδεκτές ευρετικές συναρτήσεις $h_1(n)$ και $h_4(n)$, καλύτερη είναι η $h_4(n)$ διότι δίνει καλύτερη εκτίμηση του $a(n)$ (μεγαλύτερες τιμές) και για τις δύο καταστάσεις A και B.

Ασκηση 2

Δίνεται η ακόλουθη λίστα από λέξεις:

CAB, CAD, CAT, COD, CUD, DAB, HAT, ROB, ROD, ROT

Το πρόβλημα ορίζεται ως το να μεταβούμε από μια λέξη της λίστας (αρχική κατάσταση) στη λέξη **CAT (τελική κατάσταση)** αλλάζοντας ένα μόνο γράμμα τη φορά και χρησιμοποιώντας μόνο λέξεις από την παραπάνω λίστα.

(1) Να ορίσετε τον χώρο καταστάσεων του προβλήματος, να διατυπώσετε μια **αποδεκτή** ευρετική συνάρτηση h(n) για το πρόβλημα αυτό και να αιτιολογήσετε για ποιο λόγο είναι παραδεκτή.

(2) Να σχεδιάσετε το γράφο μεταβάσεων, αναγράφοντας δίπλα σε κάθε κατάσταση την τιμή της h(n) που ορίσατε προηγουμένως.

(3) Να εκτελέσετε τον αλγόριθμο A^* για να βρείτε την μικρότερη ακολουθία λέξεων από τη λέξη ROB στη λέξη CAT.

Για την εκτέλεση του A* θα θεωρήσετε αλφαβητική προτεραιότητα μεταξύ καταστάσεων ίσου κόστους και δεν θα χρησιμοποιήσετε κλειστό σύνολο.

Στο τέλος της αναζήτησης να αναφέρετε το μονοπάτι (ακολουθία λέξεων) που βρήκατε.

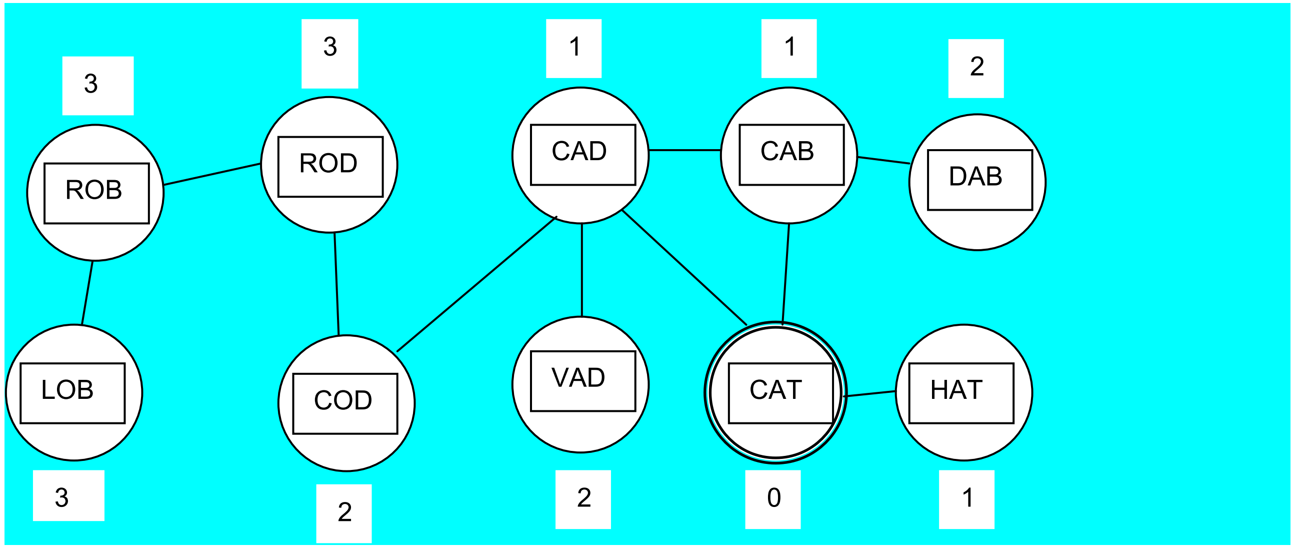
Λύση

(1) Ο χώρος καταστάσεων ορίζεται θεωρώντας ότι κάθε λέξη αποτελεί μια κατάσταση. Το κόστος κάθε μετάβασης είναι σταθερό (θεωρούμε ότι είναι ίσο με 1) και οι μεταβάσεις είναι **αμφίδρομες**.

Μια αποδεκτή ευρετική συνάρτηση $h(n)$ είναι ο αριθμός των θέσεων (από 0 έως 3) στις οποίες διαφέρει η κατάσταση *n* από την τελική κατάσταση CAT, π.χ. η λέξη CAD διαφέρει από την CAT σε μία θέση (την τελευταία), άρα h(CAD)=1. Προφανώς h(CAT)=0.

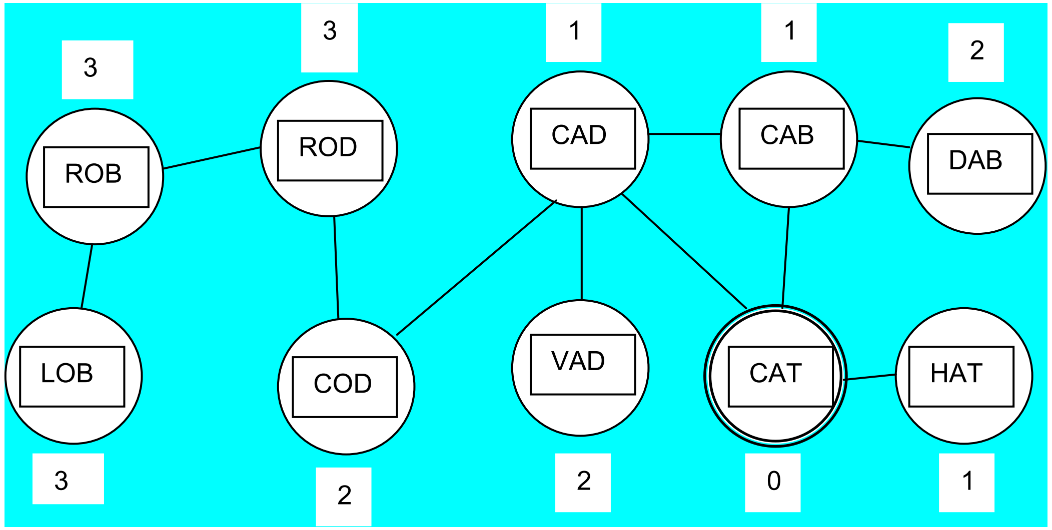
Η συνάρτηση αυτή είναι αποδεκτή διότι, αν μια κατάσταση *n* διαφέρει σε *k* θέσεις από την τελική κατάσταση CAT, θα χρειαστούν τουλάχιστον *k* μεταβάσεις για να αλλάξουν τα γράμματα σε αυτές τις *k* θέσεις (δεδομένου ότι αλλάζει ένα μόνο γράμμα σε κάθε μετάβαση).

(2) Ο γράφος μεταβάσεων είναι ο παρακάτω, όπου, δίπλα σε κάθε κόμβο, αναγράφεται η τιμή της $h(n)$.



(3) Στην **αναζήτηση A* επεκτείνεται ο κόμβος του μετώπου αναζήτησης με το μικρότερο g(n)+h(n),** όπου g(n) το κόστος μονοπατιού από την αρχική κατάσταση έως τον κόμβο n και h(n) η τιμή της ευρετικής συνάρτησης.

Η εκτέλεση της αναζήτησης έχει ως εξής:

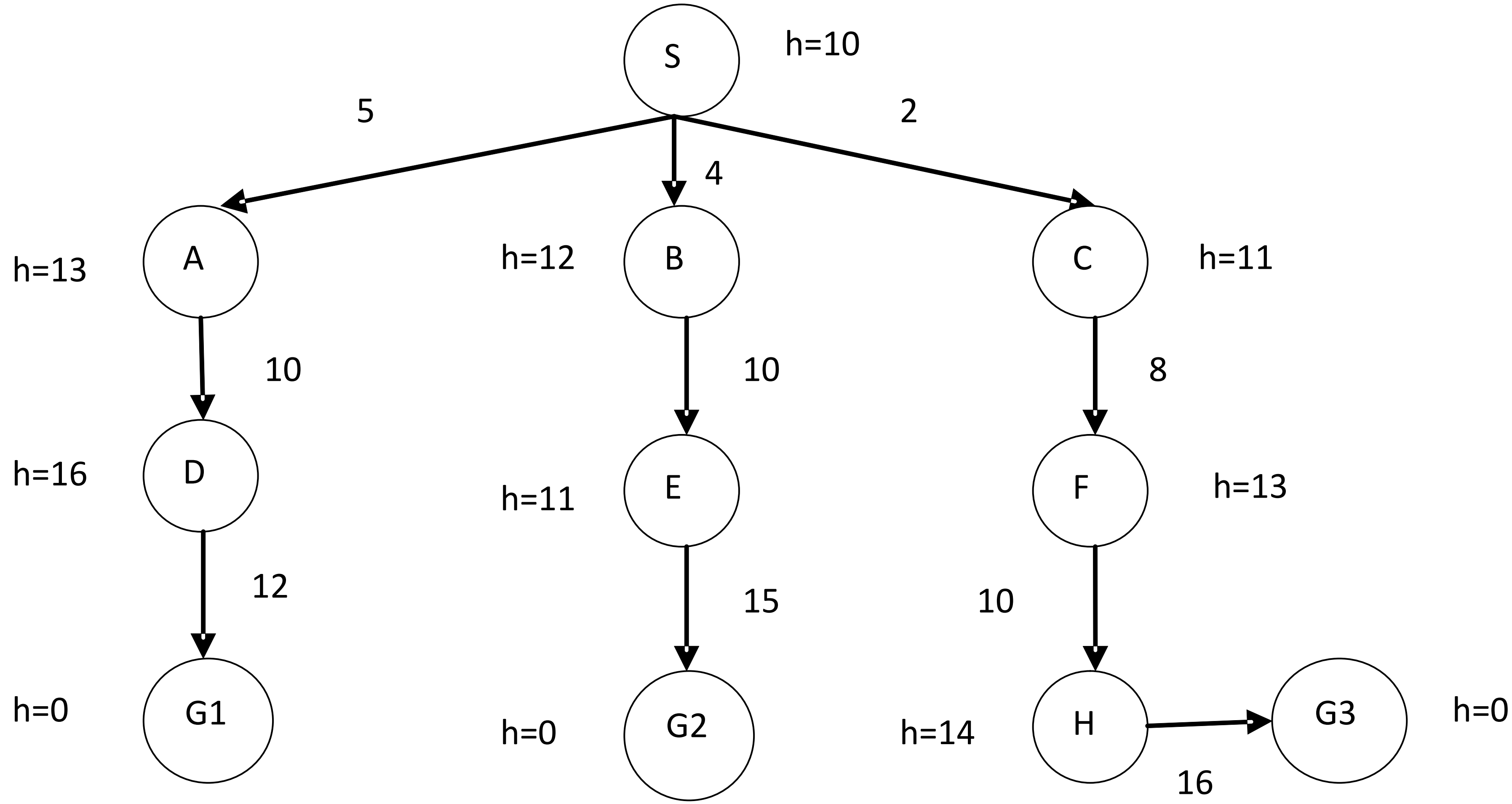


- MA={ROB(3,-)}, επιλέγεται: ROB(3,-)
- MA={ LOB(4,ROB), ROD(4,ROB)}, επιλέγεται: LOB(4,ROB)
- MA={ROD(4,ROB),ROB(5,LOB)}, επιλέγεται: ROD(4,ROB)
- MA={ROB(5,ROD),ROB(5,LOB),COD(4,ROD)}, επιλέγεται: COD(4,ROD)
- MA={ ROB(5,ROD),ROB(5,LOB),ROD(6,COD),CAD(4,COD)}, επιλέγεται: CAD(4,COD)
- MA={ROB(5,ROD),ROB(5,LOB),ROD(6,COD),COD(6,CAD),CAT(4,CAD),CAB(5,CAD) , VAD(6,CAD)}, επιλέγεται: CAT(4,CAD) που είναι τελική κατάσταση.

Η ακολουθία λέξεων (**μονοπάτι)** που βρίσκουμε είναι: **ROB**->**ROD->COD->CAD->CAT** με **κόστος 4**.

Ασκηση 3

Θεωρείστε το γράφο μεταβάσεων του παρακάτω σχήματος όπου S είναι η αρχική κατάσταση και G1, G2, G3 οι **τρεις** τελικές καταστάσεις. Πάνω στις ακμές αναγράφεται το κόστος μετάβασης μεταξύ δύο καταστάσεων και δίπλα σε κάθε κόμβο περιγράφεται το κόστος που αντιστοιχεί σε μια ευρετική συνάρτηση h(n).



Για τον παραπάνω χώρο καταστάσεων να εφαρμόσετε:

(1) Αναζήτηση A^*

(2) Αναζήτηση ομοιόμορφου κόστους.

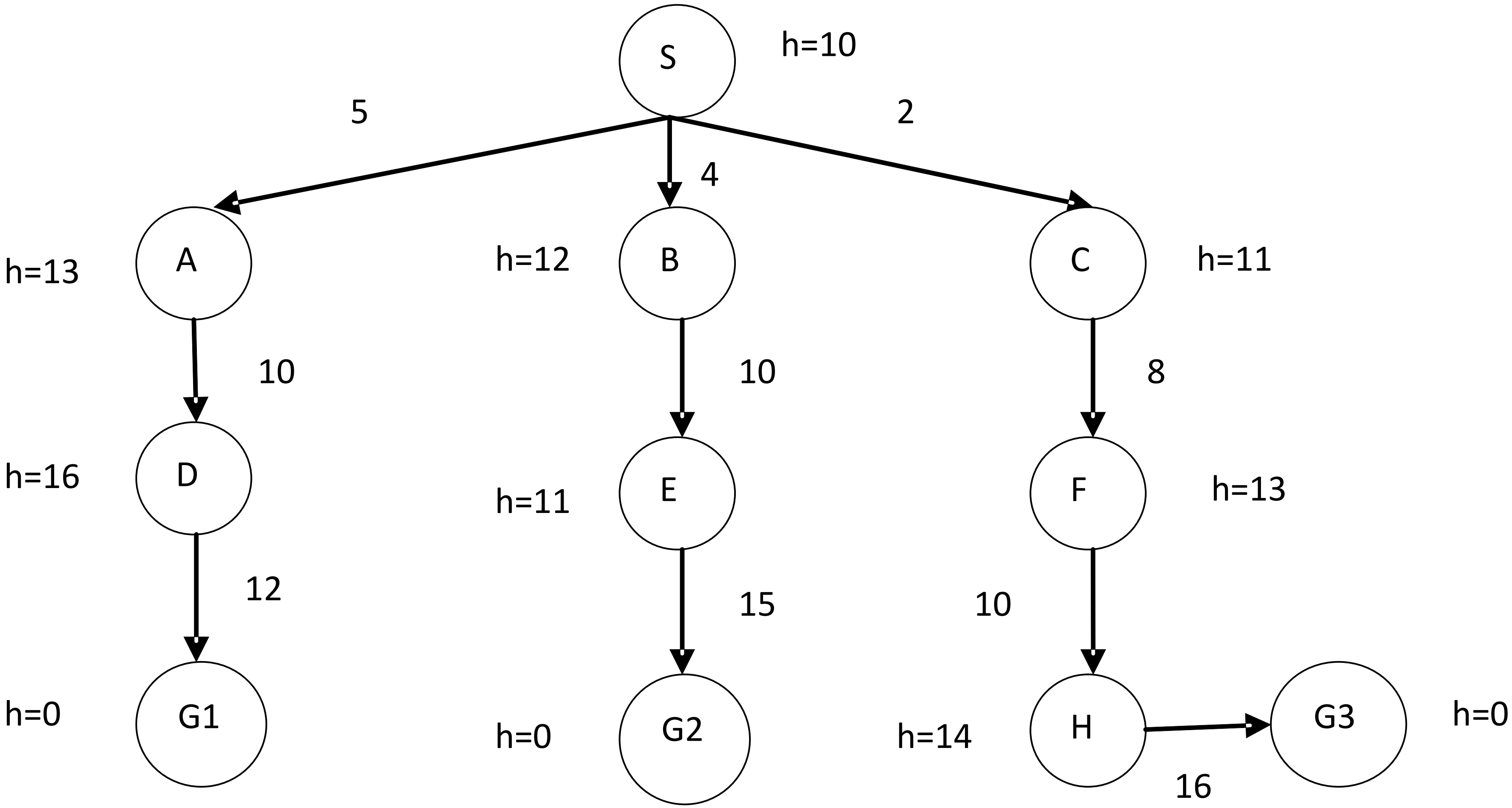
Για καθεμιά από τις παραπάνω δύο στρατηγικές αναζήτησης, να περιγράψετε την εξέλιξη της αναζήτησης αναφέροντας για κάθε βήμα τα στοιχεία του μετώπου αναζήτησης ΜΑ και την κατάσταση που επιλέγεται για επέκταση:

Στο τέλος κάθε αναζήτησης να αναφέρετε το μονοπάτι (λύση) που βρήκατε.

Σε περίπτωση καταστάσεων με ίδιο κόστος να θεωρήσετε αλφαβητική προτεραιότητα.

(3) Η αναζήτηση A^* οδηγεί στην βέλτιστη λύση; Εάν όχι, για ποιον λόγο συμβαίνει αυτό;

Λύση

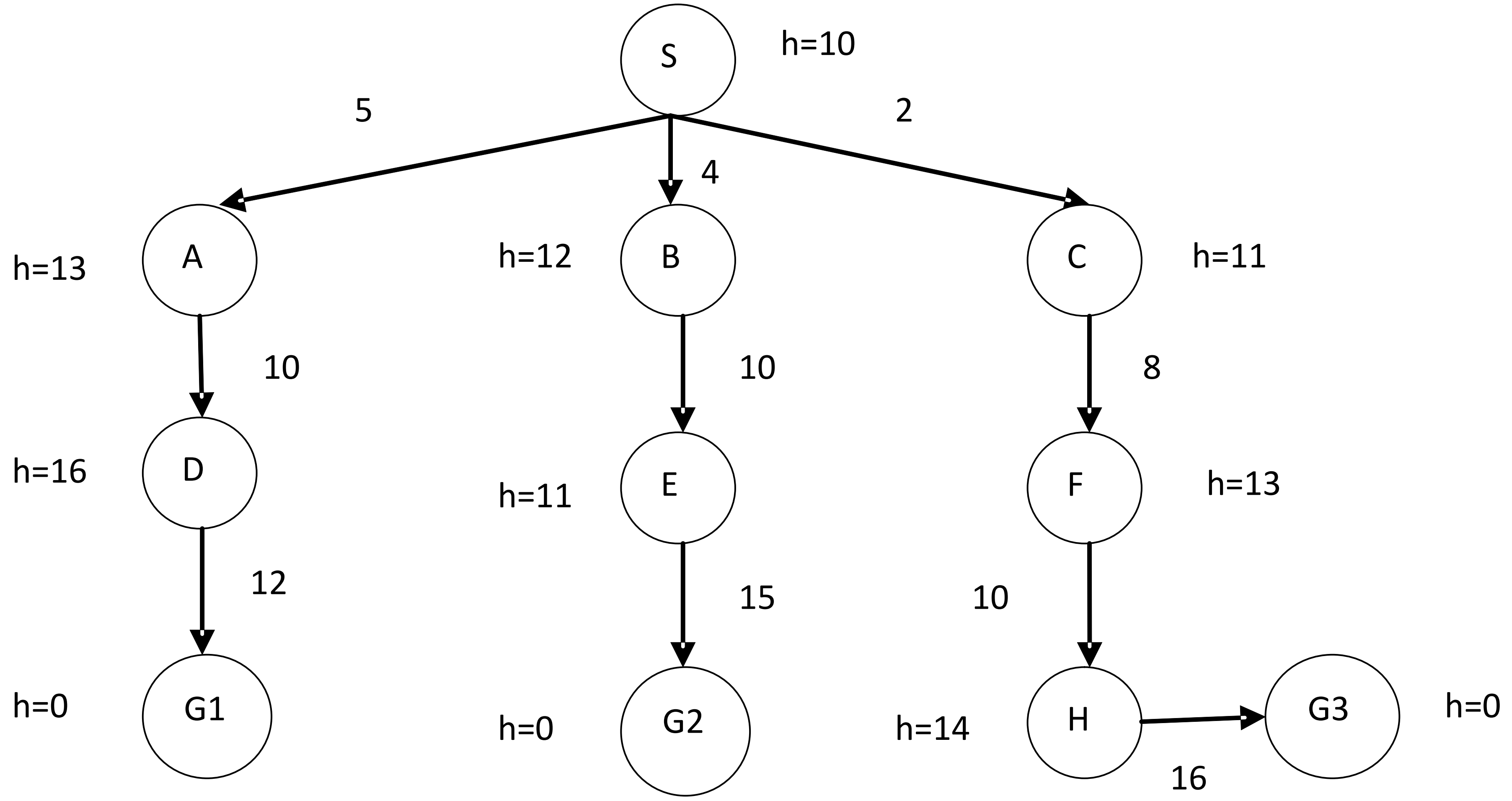


(1) Στην αναζήτηση Α* επεκτείνεται ο κόμβος του μετώπου αναζήτησης με το μικρότερο κόστος g(n)+h(n). Η εκτέλεση της αναζήτησης έχει ως εξής:

- ΜΑ={S(10,-}, επιλέγεται: S(10,-)
- ΜΑ={A(18,S),B(16,S),C(13,S)}, επιλέγεται: C(13,S)

- MA={A(18,S),B(16,S),F(23,C)}, επιλέγεται: B(16,S)
- MA={A(18,S),F(23,C),E(25,B)}, επιλέγεται: A(18,S)
- MA={F(23,C),E(25,B), D(31,A)}, επιλέγεται: F(23,C)
- MA={E(25,B), D(31,A),H(34,F)}, επιλέγεται: E(25,B)
- MA={D(31,A),H(34,F),G2(29,E)}, επιλέγεται: G2(29,E) που είναι τελική κατάσταση.

Το μονοπάτι που βρίσκουμε είναι το S -> B -> E -> G2 κόστους 29.


S
h=10
5
4
2
A
h=13
B
h=12
C
h=11
10
10
8
D
h=16
E
h=11
F
h=13
12
15
10
G1
h=0
G2
h=0
H
h=14
G3
h=0
16

(2) Στην αναζήτηση ομοιόμορφου κόστους επεκτείνεται ο κόμβος του μετώπου αναζήτησης με το μικρότερο g(n) (όπου g(n) το κόστος μονοπατιού από την αρχική κατάσταση έως τον κόμβο n). Η εκτέλεση της αναζήτησης έχει ως εξής:

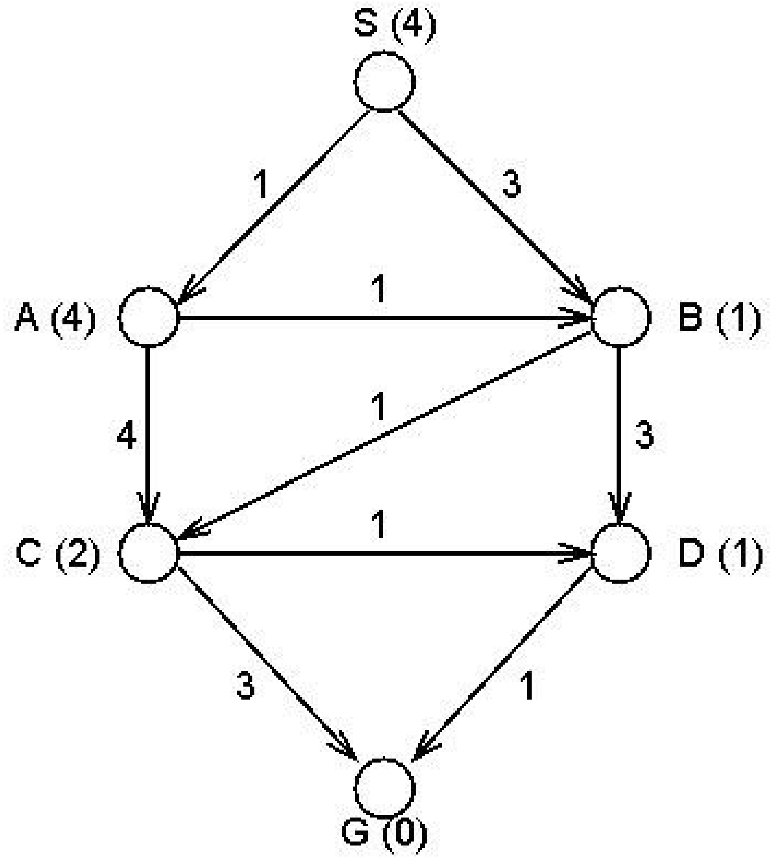
- MA={S(0,-}, επιλέγεται: S(0,-)
- MA={A(5,S),B(4,S),C(2,S)}, επιλέγεται: C(2,S)
- MA={A(5,S),B(4,S),F(10,C)}, επιλέγεται: B(4,S)
- MA={A(5,S),F(10,C),E(14,B)}, επιλέγεται: A(5,S)
- MA={F(10,C),E(14,B), D(15,A)}, επιλέγεται: F(10,C)
- MA={E(14,B), D(15,A),H(20,F)}, επιλέγεται: E(14,B)
- MA={D(15,A),H(20,F),G2(29,E)}, επιλέγεται: D(15,A)
- MA={H(20,F),G2(29,E),G1(27,D)}, επιλέγεται: H(20,F)
- MA={G2(29,E),G1(27,D),G3(36,H)}, επιλέγεται: G1(27,D) που είναι τελική κατάσταση.

Το μονοπάτι που προκύπτει είναι το S -> A ->D -> G1 κόστους 27.

(3) Παρατηρούμε ότι ενώ η μέθοδος ομοιόμορφου κόστους έδωσε τη βέλτιστη λύση, η αναζήτηση A* δεν οδήγησε στη βέλτιστη λύση. Αυτό οφείλεται στο ότι **η h(n) που χρησιμοποιήθηκε δεν είναι αποδεκτή**. Για παράδειγμα h(D)=16, ενώ το κόστος του D από την τελική κατάσταση G1 είναι 12.

Ασκηση 4

Θεωρήστε τον γράφο μεταβάσεων του παρακάτω σχήματος. Το ζητούμενο είναι να μεταβούμε από τον αρχικό κόμβο S στον κόμβο-στόχο G. Οι αριθμοί δίπλα στις ακμές, είναι τα κόστη μετάβασης μεταξύ των κόμβων. Ο αριθμός δίπλα σε ένα κόμβο είναι η τιμή μίας ευρετικής συνάρτησης που επιστρέφει μία εκτίμηση της απόστασης του κόμβου από τον κόμβο-στόχο.



(1) Είναι η ευρετική συνάρτηση αποδεκτή ή όχι και γιατί;

(2) Εάν εκτελέσετε i) αναζήτηση A* με την παραπάνω h(n) και ii) αναζήτηση ομοιόμορφου κόστους, το κόστος της λύσης θα είναι το ίδιο στις δύο περιπτώσεις;

Λύση:

(1) Στον παρακάτω πίνακα παρουσιάζονται για κάθε κόμβο n i) το μήκος της ελάχιστης απόστασης α(n) προς τον κόμβο-στόχο G και ii) η αντίστοιχη τιμή της ευρετικής συνάρτησης h(n).

	S	A	B	C	D	G
α(n)	5	4	3	2	1	0
h(n)	4	4	1	2	1	0

Παρατηρούμε ότι σε καμία περίπτωση η τιμή της ευρετικής συνάρτησης δεν υπερεκτιμά το βέλτιστο κόστος. Άρα η ευρετική συνάρτηση είναι αποδεκτή.

(2) Αφού η ευρετική h(n) είναι αποδεκτή η αναζήτηση A* είναι βέλτιστη. Το ίδιο ισχύει και για την αναζήτηση ομοιόμορφου κόστους. Αρα και οι δύο μέθοδοι θα δώσουν τη λύση με ίδιο ελάχιστο κόστος.

Ασκηση

Έστω ότι εφαρμόζουμε τον γενικό αλγόριθμο αναζήτησης, αλλά σε κάθε βήμα επεκτείνεται ο κόμβος n του μετώπου αναζήτησης με το **μικρότερο**

$$f(n)=w\, g(n) + (1-w)\, h(n), \text{ όπου } 0 \le w \le 1,$$

h(n) είναι **αποδεκτή** ευρετική συνάρτηση και g(n) το κόστος μονοπατιού από την αρχική κατάσταση έως τον κόμβο n.

Να αιτιολογήσετε εάν η παραπάνω μέθοδος είναι βέλτιστη για:

i) $w=0$, ii) $w=1$, iii) $w=0.5$.

Λύση

i) w=0, f(n)=h(n), άπληστη αναζήτηση, δεν είναι βέλτιστη

ii) w=1, f(n)=g(n), αναζήτηση ομοιόμοροφυ κόστους, βέλτιστη

iii) w=0.5, f(n)=0.5*(g(n)+h(n)), **ισοδύναμη** (γιατί;) με A*, βέλτιστη

Ασκηση

Έστω δύο **αποδεκτές** ευρετικές συναρτήσεις $h_1(n)$ και $h_2(n)$ για κάποιο πρόβλημα αναζήτησης.

Να δείξετε ότι η συνάρτηση $h_3(n)=(h_1(n)+h_2(n))/2$ είναι αποδεκτή ευρετική συνάρτηση.

Λύση

Αφού h1(n), h2(n), αποδεκτές, ισχύει για κάθε n:

$h_1(n) \le \alpha(n) \rightarrow 0.5\ h_1(n) \le 0.5\ \alpha(n)$

$h_2(n) \le \alpha\ (n) \rightarrow 0.5\ h_2(n) \le 0.5\ \alpha(n)$

Προσθέτοντας κατά μέλη προκύπτει ότι $h_3(n) \le \alpha(n)$, άρα $h_3(n)$ αποδεκτή.