Τεχνητή Νοημοσύνη
1η Αναλυτική Εργασία

Ioannis Daras (daras.giannhs@gmail.com)
Αριθμός Μητρώου: 03115018
Ροές: Υ, Λ, Σ

"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it." —Eliezer Yudkowsky

---

Άσκηση 1

Hill Climbing

Ο αλγόριθμος Hill Climbing είναι ένας greedy αλγόριθμος αναζήτησης ο οποίος όταν εξετάζει μια κατάσταση ταξινομεί τις καταστάσεις παιδιά της με βάση την τιμή ενός heuristic function και βάζει στο ανοικτό μέτωπο την κατάσταση που η heuristic function θεωρεί καλύτερη μόνο αν η τιμή της συνάρτησης για την κατάσταση παιδί είναι καλύτερη από την τιμή της συνάρτησης για την τρέχουσα κατάσταση. Σε διαφορετική περίπτωση, ο αλγόριθμος σταματάει.

Η εκτέλεση του αλγορίθμου για το γράφο της εκφώνησης φαίνεται σε μορφή πίνακα παρακάτω:

Open set	Closed set	Current state	Children states	Heuristic
[$(s,9)^{s}$]	n/a	s	[b,c,d]	[(b,6), (c,4), (d,5)]
[$(c,4)^{s,c}$]	n/a	c	[h]	[(h,5)]

Best First

Ο αλγόριθμος Best First τοποθετεί όποια κατάσταση δεν έχει ξαναδεί στο ανοικτό σύνολο αλλά κάθε φορά το ανοικτό σύνολο το έχει διατεταγμένο ως προς τις τιμές που δίνει το heuristic για κάθε κατάσταση που έχει συναντήσει. Έτσι, κάνει expand την κατάσταση που φαίνεται περισσότερο ελπιδοφόρα από όσες έχει ήδη συναντήσει.

Η εκτέλεση του αλγορίθμου για το γράφο της εκφώνησης φαίνεται παρακάτω:

Open set	Closed set	Current state	Children states	Heuristic
[$(s,9)^{s}$]	[]	s	[b,c,d]	[(b,6), (c,4), (d,5)]
[$(c,4)^{s,c}$, $(d,5)^{s,d}$, $(b,6)^{s,b}$]	[s]	c	[h]	[(h,5)]
[$(d,5)^{s,d}$, $(h,5)^{s,c,h}$, $(b,6)^{s,b}$]	[s, c]	d	[h, i]	[(h,5), (i,3)]
[$(i,3)^{s,d,i}$, $(h,5)^{s,d,h}$, $(b,6)^{s,b}$]	[s, c, d]	i	[j]	[(j,3)]
[$(j,3)^{s,d,i,j}$, $(h,5)^{s,d,h}$, $(b,6)^{s,b}$]	[s, c, d, i]	j	[g]	[(g,0)]
[$(g,0)^{s,d,i,j,g}$, $(h,5)^{s,d,h}$, $(b,6)^{s,b}$]	[s, c, d, i, j]	g	[]	[]

A* Star

Ο αλγόριθμος A* Star λειτουργεί με τρόπο που μοιάζει με τον αλγόριθμο Best First μόνο που το ανοικτό μέτωπο είναι ένα διατεταγμένο σύνολο ως προς άλλη μετρική. Η μετρική εδώ είναι ένας γραμμικός συνδυασμός των τιμών που έχει δώσει το heuristic για την κάθε κατάσταση ($h(n)$) και του κόστους που πραγματικά έχει δαπανηθεί ώστε να φτάσουμε σε αυτή την κατάσταση $g(n)$. Έτσι, προκύπτει η μετρική:

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

με βάση την οποία διατάσσουμε το ανοικτό σύνολο κάθε φορά που εξετάζεται κάποια κατάσταση.

Η εκτέλεση του αλγορίθμου για τον γράφο της εκφώνησης είναι η ακόλουθη:

Open set	Closed set	Current state	Children states	G	F
[$(s,0,9)^{s}$]	[]	s	[b,c,d]	[(b,5), (c,2), (d,2)]	[(b,11), (c,6), (d,7)]
[$(c,2,6)^{s,c}$, $(d,2,7)^{s,d}$, $(b,5,11)^{s,b}$]	[s]	c	[h]	[(h,8)]	[(h,13)]
[$(d,2,7)^{s,d}$, $(b,5,11)^{s,b}$, $(h,8,13)^{s,c,h}$]	[s, c]	d	[h, i]	[(h,10), (i,8)]	[(h,15), (i,11)]
[$(b,5,11)^{s,b}$, $(i,8,11)^{s,d,i}$, $(h,8,13)^{s,c,h}$]	[s, c, d]	b	[k,e]	[(k,7), (e,9)]	[(k,9), (e,14)]
[$(k,7,9)^{s,b,k}$, $(i,8,11)^{s,d,i}$, $(h,8,13)^{s,c,h}$, $(e,9,\ 14)^{s,b,e}$]	[s,c,d,b]	k	[g, h]	[(g,18), (h,8)]	[(g,18), (h,13)]
[$(i,8,11)^{s,d,i}$, $(h,8,13)^{s,c,h}$, $(e,9,14)^{s,b,e}$, $(g,\ 18,18)^{s,b,k,g}$]	[s,c,d,b,k]	i	[j]	[(j,12)]	[(j, 15)]
[$(h,8,13)^{s,c,h}$, $(e,9,14)^{s,b,e}$, $(j,12,15)^{s,d,i,j}$, $(g,18,18)^{s,b,k,g}$]	[s,c,d,b,k,i]	h	[j]	[(j,15)]	[(j,18)]
[$(e,9,14)^{s,b,e}$, $(j,12,15)^{s,d,i,j}$, $(g,18,18)^{s,b,k,g}$]	[s,c,d,b,k,i,h]	e	[g]	[(g,18)]	[(g,18)]
[$(j,12,15)^{s,d,i,j}$, $(g,18,18)^{s,b,k,g}$]	[s,c,d,b,k,i,h,e]	j	[g]	[(g,14)]	[(g,14)]
[$(g,14,14)^{s,d,i,j,g}$]	[s,c,d,b,k,i,h,e,j]	g	[]	[]	[]

Ανάλυση προβλήματος

Αρχικά, το γράφημα δεν έχει κύκλους άρα ο αριθμός των λύσεων θα είναι πεπερασμένος. Συγκεκριμένα, έχουμε 9 λύσεις οι οποίες είναι οι ακόλουθες ακολουθίες κορυφών:

$$[s, b, e, g]$$

$$[s, b, k, g]$$

$$[s, c, h, j, g]$$

$$[s, d, h, j, g]$$

$$[s, d, h, i, j, g]$$

$$[s, b, k, h, j, g]$$

$$[s, b, k, h, i, j, g]$$

$$[s, d, i, j, g]$$

$$[s, c, h, i, j, g]$$

Η καλύτερη ακολουθία είναι αυτή που βρίσκουν ο Best First και ο A* Star και είναι η:

$$[s, d, i, j, g]$$

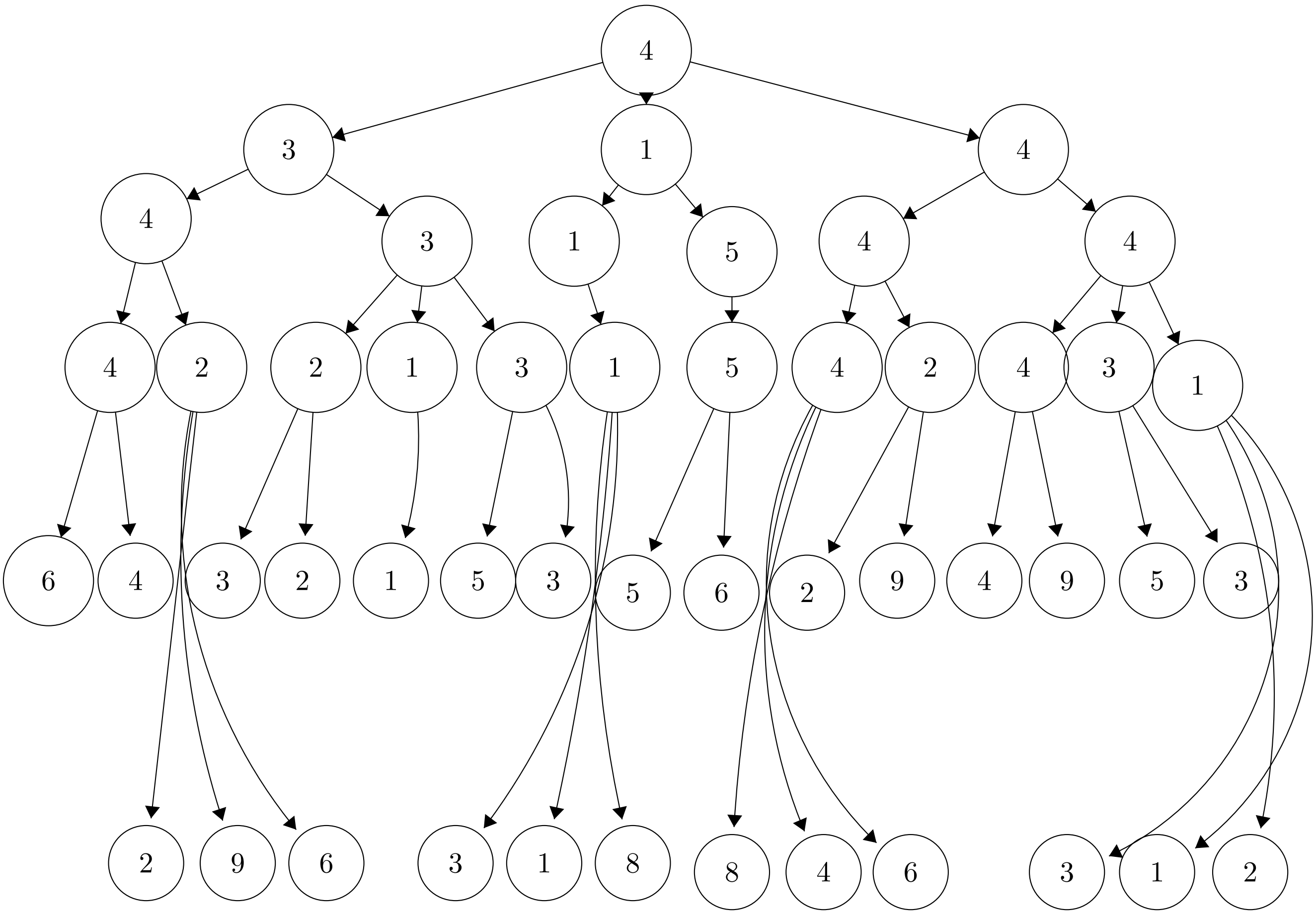
Ο Hill Climbing δεν βρίσκει καμία λύση. Γενικά, ο Hill Climbing αλγόριθμος δίνει λύσεις σε convex προβλήματα, υποσύνολο των οποίων είναι τα προβλήματα με ένα ακρότατο. Εδώ έχουμε περισσότερα από ένα ακρότατα οπότε ο Hill Climbing δεν μπορεί να λύσει το πρόβλημα. Για τον Best First δεν υπάρχει κάποια θεωρητική θεμελίωση για το λόγο που βρίσκει τη σωστή λύση στο συγκεκριμένο πρόβλημα, το ότι βρίσκει τη βέλτιστη λύση εξαρτάται καθαρά δηλαδή από το instance του προβλήματος. Για τον A* γνωρίζουμε ότι βρίσκει εγγυημένα τη βέλτιστη λύση αν το heuristic function δεν υπερεκτιμά ποτέ την απόσταση ενός κόμβου από τον κόμβο προορισμού. Εδώ, συμβαίνει 1 υπερεκτίμηση, στον κόμβο j, παρόλα αυτά, το g έχει ήδη μπει στο ανοικτό μέτωπο και έτσι βρίσκεται και πάλι η βέλτιστη λύση στο συγκεκριμένο πρόβλημα.

Άσκηση 2

Minimax

Ο αλγόριθμος Minimax είναι ένας αλγόριθμος που επιχειρεί να βρει τη στρατηγική νίκης σε ένα παιχίδι δύο αντιπάλων που παίζουν ο ένας μετά τον άλλον εναλλάξ. Κάθε κατάσταση αξιολογείται με έναν αριθμό που δείνει ένα heuristic function. Αν αυτός ο αριθμός είναι πάνω από ένα threshold τότε κερδίζει ο παίκτης A ενώ αν είναι κάτω από ένα threshold κερδίζει ο παίκτης B. Έτσι, ο παίκτης A προσπαθεί να μεγιστοποιήσει τον αριθμό της κατάστασης στην οποία βρίσκεται διαλέγοντας από τις πιθανές του κινήσεις εκείνη που του δίνει τον μεγαλύτερο αριθμό. Αντίθετα, ο παίκτης B προσπαθεί να ελαχιστοποιήσει τον αριθμό της κατάστασης στην οποία βρίσκεται διαλέγοντας από τις πιθανές κινήσεις του εκείνη που θα δώσει τον ελάχιστο αριθμό.

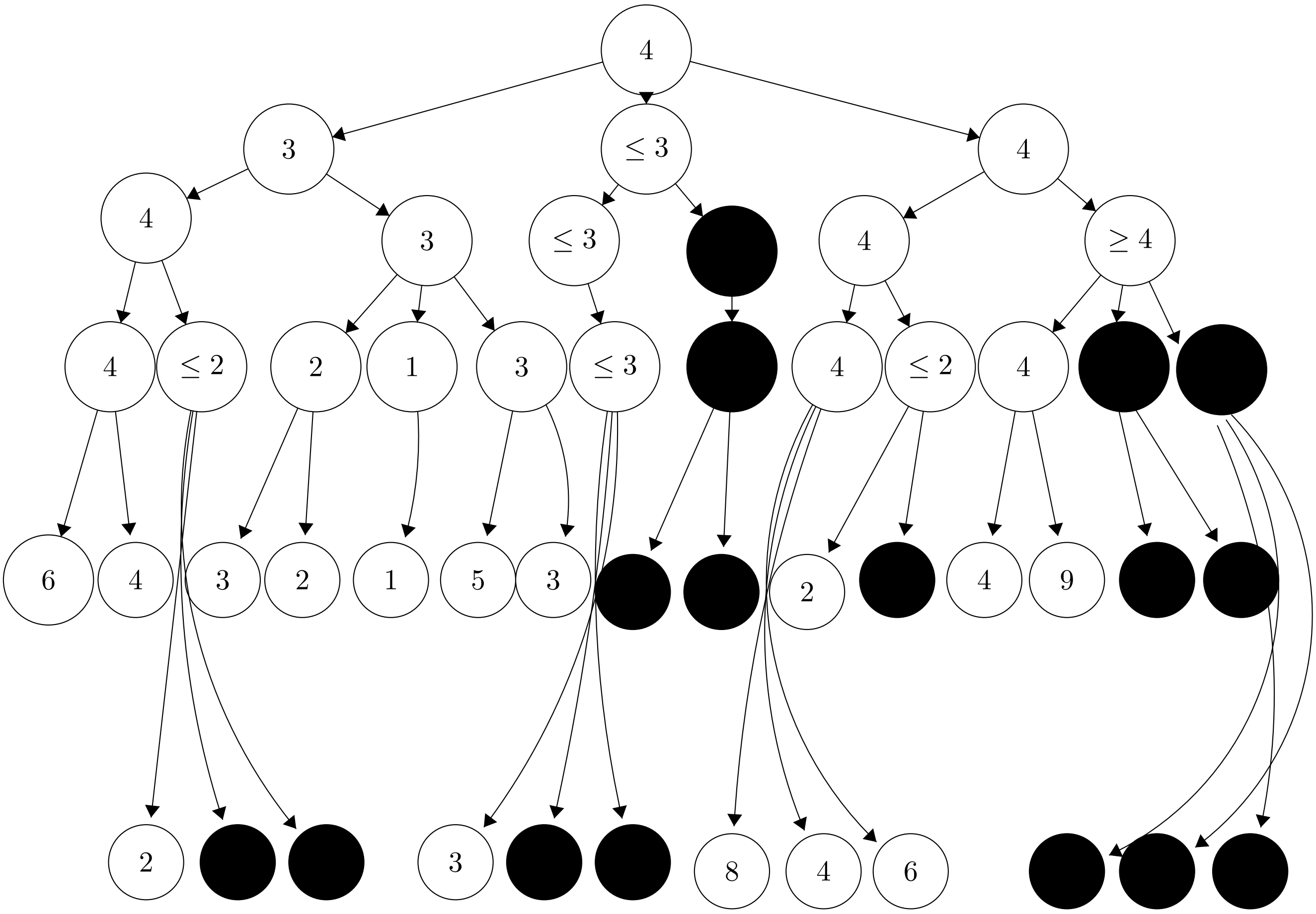
Η εκτέλεση του αλγορίθμου για το γράφο της εκφώνησης φαίνεται παρακάτω:



AB Pruning

Ο AB Pruning αποτελεί μια βελτίωση του αλγορίθμου Minimax. Σε μια πραγματική εφαρμογή του Minimax ο υπολογισμός των τιμών του heuristic πάνω στα φύλλα είναι η πραγματικά χρονοβόρα ενέργεια αφού οι τιμές των φύλλων καθορίζουν τις τιμές των υπολοίπων κόμβων. Ο AB Pruning αλγόριθμος πρόκυψε από την παρατήρηση ότι ορισμένες τιμές φύλλων (και άλλων κόμβων) δεν χρειάζεται να υπολογιστούν καθώς οι στρατηγικές των παικτών (Max για τον A και Min για τον B) εισάγουν περιορισμούς που καθιστούν αδυνάτο να επηρεάζουν οι συγκεκριμένοι κόμβοι το αποτέλεσμα και συνεπώς ο υπολογισμός τους είναι περιττός.

Παρακάτω, φαίνεται η εκτέλεση του AB Pruning αλγορίθμου για τον γράφο της εκφώνησης:



Η σειρά με την οποία επισκέπτεται ο αλγόριθμος AB Pruning τους κόμβους είναι η ακόλουθη:

$$1, 2, 5, 11, 23, 24, 12, 25, 6, 13, 28, 29, 14, 30, 15, 31, 32, 3, 7, 16, 33, 4, 9, 18, 38, 39, 40, 19, 41, 10, 20, 43, 44$$

Αν δεν μας ενδιαφέρει η σειρά επίσκεψης αλλά η σειρά αποτίμησης των κόμβων, τότε η ζητούμενη σειρά είναι η:

$$23, 24, 11, 25, 12, 5, 28, 29, 13, 30, 14, 31, 32, 15, 6, 2, 33, 35, 16, 7, 3, 38, 39, 40, 18, 41, 19, 9, 43, 44, 20, 10, 4, 1$$

Ανάλυση παιχνιδιού

Ονομάζουμε τον παίκτη του Max επιπέδου A και τον παίκτη του Min επιπέδου B. Πρώτος παίζει ο A και παίζει την κίνηση που οδηγεί στον κόμβο 4. Στη συνέχεια, ο παίκτης B αν εφαρμόζει αλγόριθμο Minimax μπορεί να παίξει οποιαδήποτε εκ των κινήσεων που οδηγούν στους κόμβους 9, 10 ενώ αν εφαρμόζει AB Pruning τότε θα παίξει την κίνηση που οδηγεί στον κόμβο 9. Στην περίπτωση που παίξει την κίνηση που οδηγεί στον κόμβο 9 τότε ο A θα παίξει μετά την κίνηση για τον κόμβο 18 και ο B θα παίξει συνέχεια την κίνηση για τον κόμβο 39. Στην περίπτωση που παίξει την κίνηση που οδηγεί στον κόμβο 10 τότε ο A θα παίξει την κίνηση που οδηγεί στον κόμβο 20 και στη συνέχεια ο B θα παίξει την κίνηση που οδηγεί στον κόμβο 43. Έτσι, οι πιθανές ακολουθίες κινήσεων είναι:

$$1, 4, 9, 18, 39$$

$$1, 4, 10, 20, 43$$