ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Τεχνητή νοημοσύνη και μηχανική μάθηση

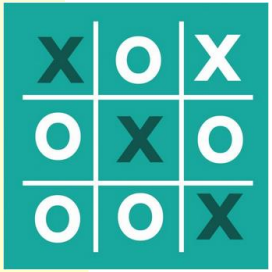
Εισηγητής
Αναστάσιος Κεσίδης

Αναζήτηση με αντιπαλότητα

Αναζήτηση με αντιπαλότητα

Γενικά

Τα **ανταγωνιστικά παιχνίδια** (**adversarial games**) ή παιχνίδια **δύο αντιπάλων** (two-person games) είναι προβλήματα αναζήτησης λύσης όπου η εξέλιξη των καταστάσεων εξαρτάται από δύο **διαφορετικά** σύνολα τελεστών μετάβασης που εφαρμόζονται **διαδοχικά**.



Αναζήτηση με αντιπαλότητα

Γενικά

Ορισμός του προβλήματος

Μια **κατάσταση** παριστάνει τη διάταξη των αντικειμένων του παιχνιδιού (π.χ. πιόνια) σε κάποια χρονική στιγμή.

Ο χώρος καταστάσεων αποτελείται από **όλες τις έγκυρες καταστάσεις**.

Οι **τελεστές μετάβασης** είναι οι επιτρεπτές κινήσεις σε κάθε κατάσταση που καθορίζονται από τους κανόνες του παιχνιδιού.

Οι **τελικές καταστάσεις** του παιχνιδιού καθορίζονται από κριτήρια (π.χ. ματ στο σκάκι)

Παίκτες

Δύο παίκτες που παίζουν **εναλλάξ**.

Ο ένας παίκτης λέγεται **MAX**

Ο άλλος παίκτης λέγεται **MIN**

Αναζήτηση με αντιπαλότητα

Διαδικασία επίλυσης

Ορίζεται μια **συνάρτηση αξίας** που αποδίδει μια τιμή σε κάθε τελική κατάσταση του παιχνιδιού

Η συνάρτηση αξίας πρέπει να είναι τέτοια ώστε

- Όταν κερδίζει ο παίκτης MAX η συνάρτηση αξίας **μεγιστοποιείται**
- Όταν κερδίζει ο παίκτης MIN η συνάρτηση αξίας **ελαχιστοποιείται**

Συνεπώς,

- ο MAX παίζει για να οδηγήσει το παιχνίδι σε **τελική κατάσταση με μεγάλη αξία**
- ο MIN παίζει ώστε να οδηγήσει το παιχνίδι σε **τελική κατάσταση με μικρή αξία**

Αναζήτηση με αντιπαλότητα

➢ Συνάρτηση αξίας

Καθορίζει την τελική **αριθμητική τιμή** για ένα παιχνίδι που τελειώνει σε τερματική κατάσταση για έναν από τους παίκτες.

π.χ. στο σκάκι, το αποτέλεσμα είναι νίκη, ήττα ή ισοπαλία, με τιμές +1, 0 ή ½

Παιχνίδι **μηδενικού (ή σταθερού) αθροίσματος** ορίζεται το παιχνίδι όπου η συνολική απόδοση σε όλους τους παίκτες είναι ίδια (σταθερή) σε κάθε εκτέλεση του παιχνιδιού.

πχ. το σκάκι είναι παιχνίδι μηδενικού αθροίσματος διότι κάθε παιχνίδι έχει τελική τιμή $0 + 1$ ή $1 + 0$ ή $½ + ½$.

Παραδοχές – απλοποιήσεις

- Πλήρως **γνωστό περιβάλλον** και **συνέπεια** κινήσεων.
- Δεν υπάρχει **τυχαιότητα** (π.χ. ζάρια).

Αναζήτηση με αντιπαλότητα

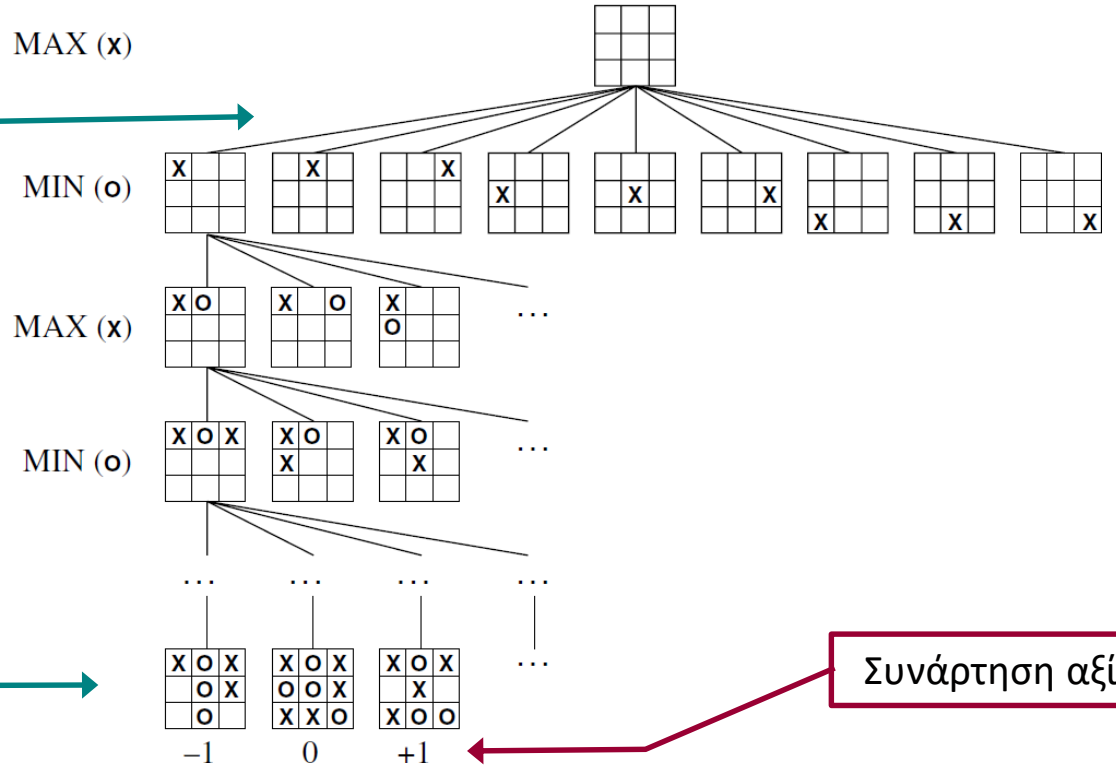
➢ Δέντρο παιχνιδιού

Το δένδρο που περιγράφει όλες τις δυνατές ακολουθίες ενεργειών ονομάζεται **δένδρο παιχνιδιού** (**game tree**).

Χαρακτηριστικό

Οι κινήσεις δύο **διαδοχικών** επιπέδων ανήκουν σε **διαφορετικούς παίκτες** καθώς αυτοί παίζουν **εναλλάξ**

Ο παίκτης **MAX** παίζει πρώτος τοποθετώντας το σύμβολο X σε κάποια από τις 9 θέσεις



Τελική κατάσταση

Συνάρτηση αξίας

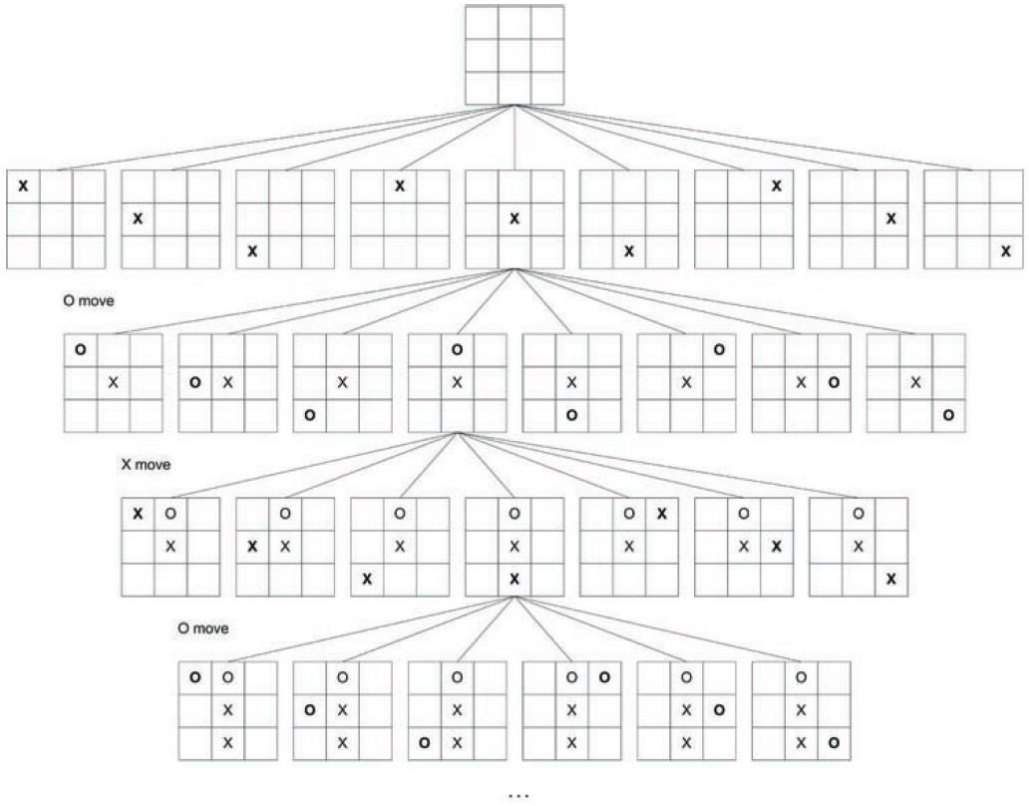
Αναζήτηση με αντιπαλότητα

Πολυπλοκότητα

Το σύνολο των καταστάσεων του παιχνιδιού μπορεί να είναι τεράστιο.

Παράδειγμα 1: Τρίλιζα

$9 \cdot 8 \cdot 7 \ldots = 9! = 362880$ καταστάσεις



Αναζήτηση με αντιπαλότητα

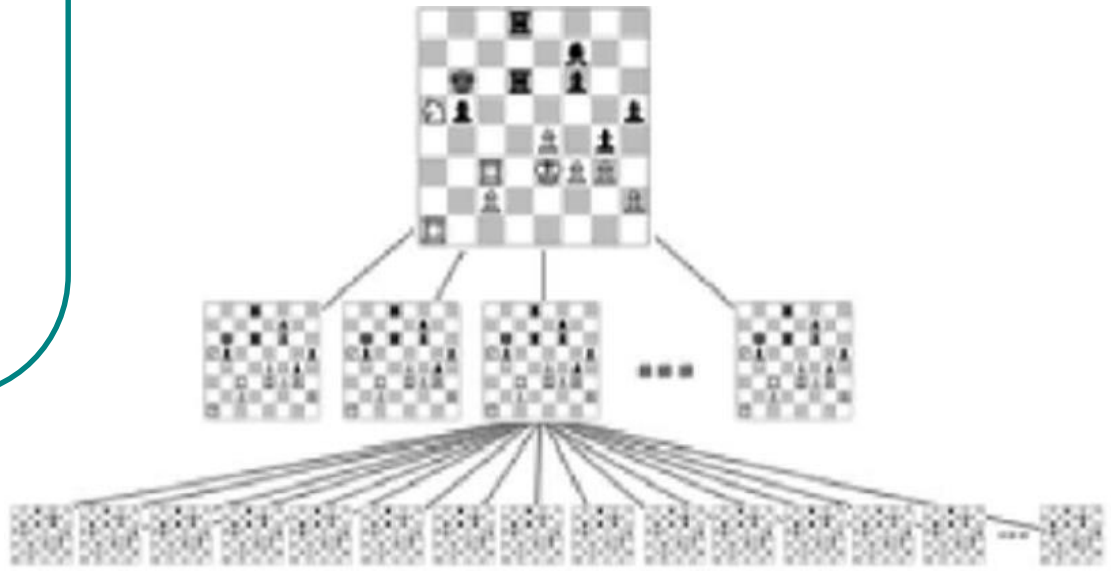
- Πολυπλοκότητα (συν.)

Παράδειγμα 2: Σκάκι

Συντελεστής διακλάδωσης $\boldsymbol{b}$
(διαθέσιμες κάθε φορά κινήσεις):
Κατά μέσο όρο 35

Βάθος δένδρου $\boldsymbol{m}$
(συνήθως 40 διαδοχικές κινήσεις
για κάθε παίκτη)

Συνεπώς:
Δένδρο αναζήτησης: $\boldsymbol{b^m} \sim 35^{40}$
κόμβοι



Minimax

Αλγόριθμος

Πλήρης **ανάπτυξη όλου του δένδρου** παιχνιδιού και υπολογισμός της **συνάρτησης αξίας** για όλες τις τελικές καταστάσεις (**τελικοί κόμβοι**) του δένδρου

Υπολογισμός όλων των **ενδιάμεσων** κόμβων, εκκινώντας από το **τέλος** του δένδρου (τελικές καταστάσεις) και με **ανάστροφη πορεία** προς την ρίζα, ως εξής:

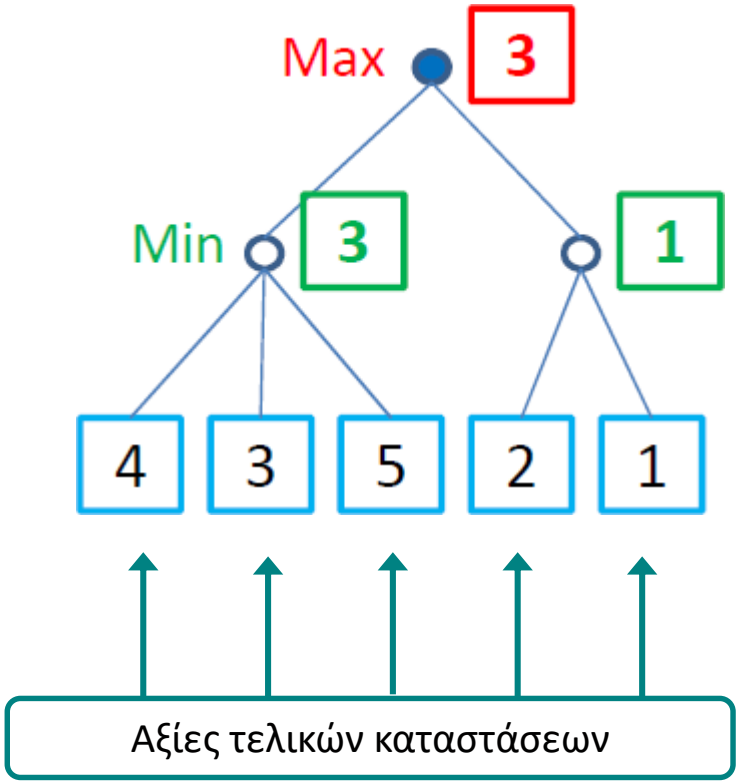
- η αξία κάθε MAX κόμβου είναι η **μέγιστη** από τις αξίες των κόμβων απογόνων του
- η αξία κάθε MIN κόμβου είναι η **ελάχιστη** από τις αξίες των κόμβων απογόνων του

Ολοκλήρωση: όταν υπολογιστεί η αξία για **τον κόμβο-ρίζα**.

Στόχος του MAX: η τιμή +1
Στόχος του MIN: η τιμή -1

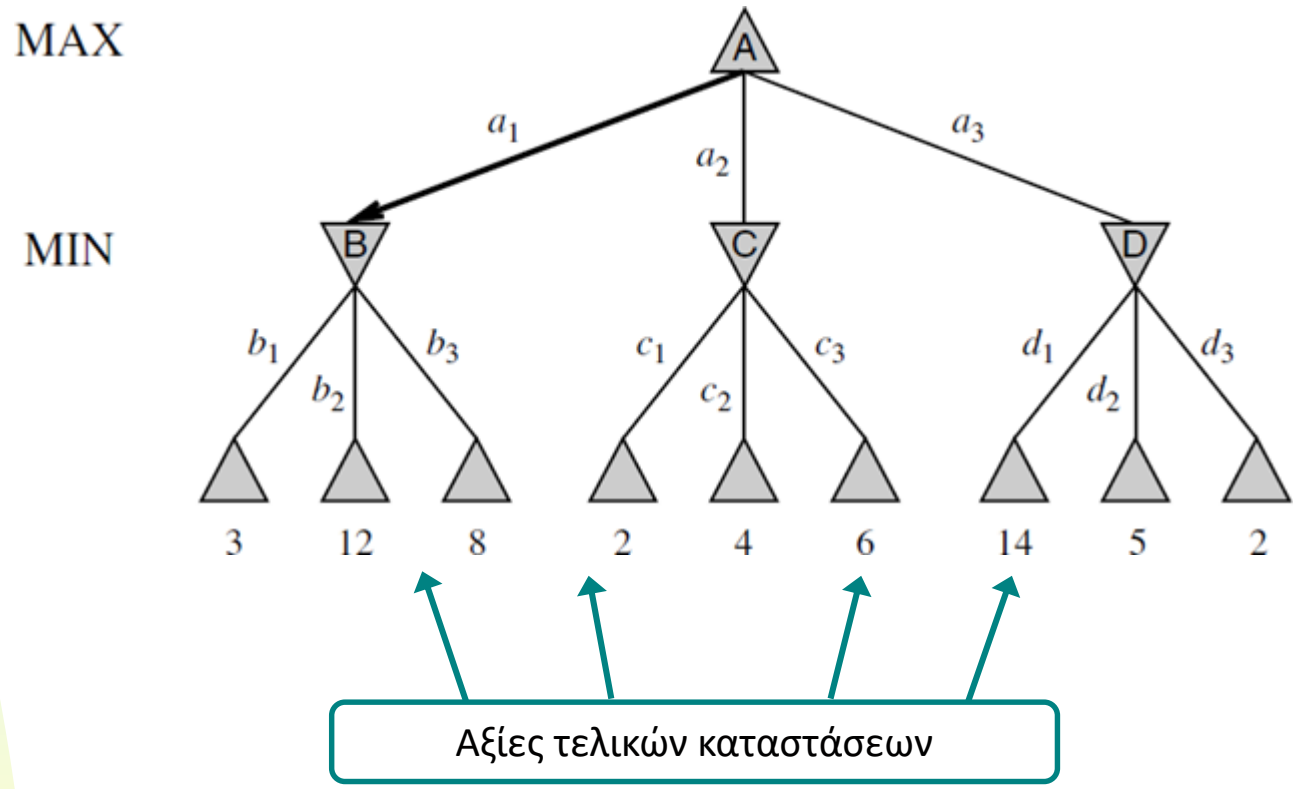
Minimax

- Παράδειγμα 1



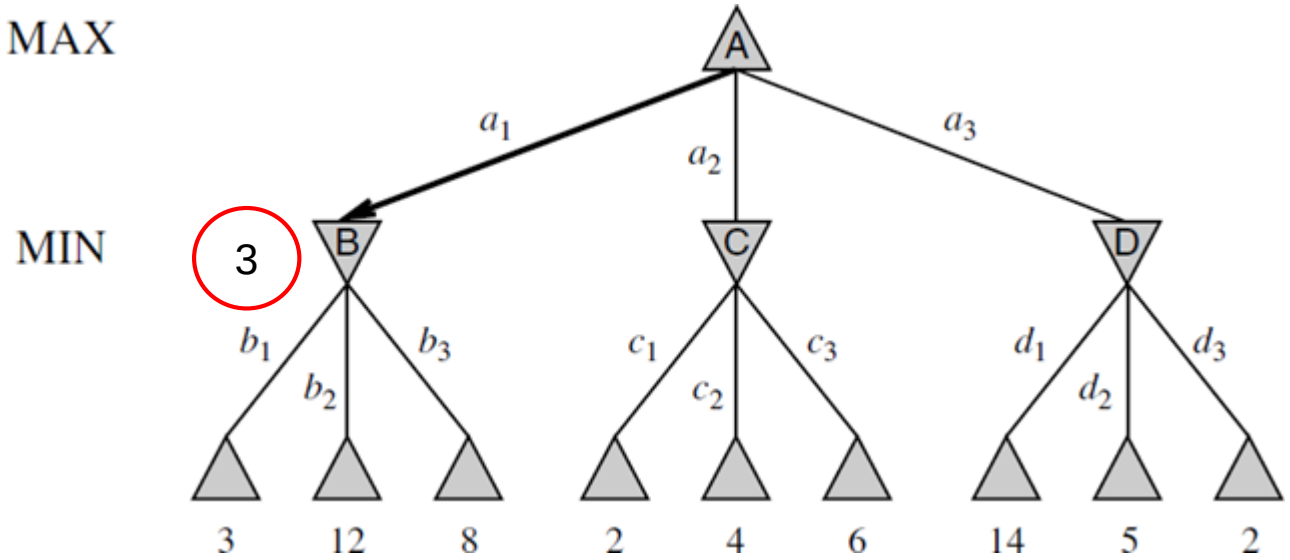
Minimax

- Παράδειγμα 2



Minimax

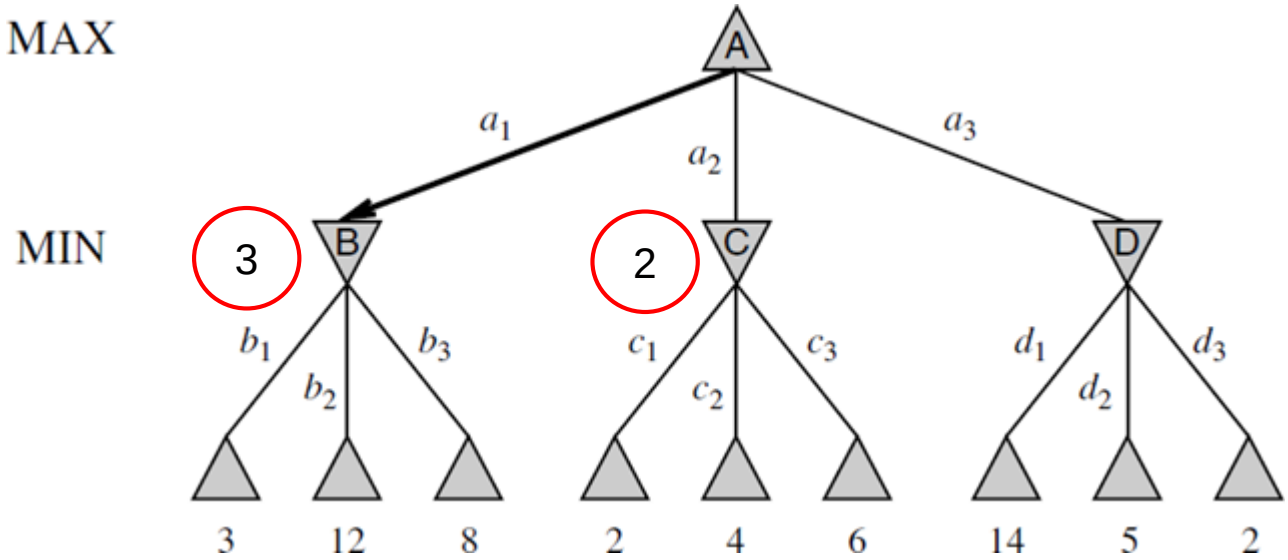
- Παράδειγμα 2 (συν.)



Ο κόμβος B είναι στην σειρά του MIN. Συνεπώς, επιλέγεται η πιο **μικρή** από τις αξίες των απογόνων του, δηλαδή η τιμή 3.

Minimax

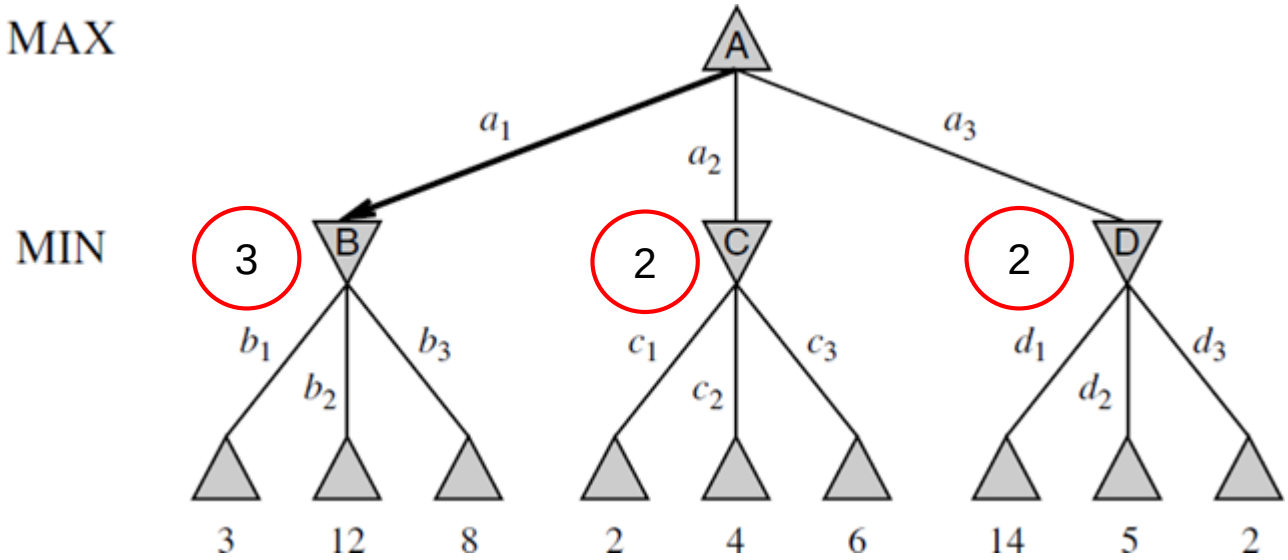
- Παράδειγμα 2 (συν.)



Ο κόμβος C είναι επίσης στην σειρά του MIN. Συνεπώς, επιλέγεται η πιο **μικρή** από τις αξίες των απογόνων του, δηλαδή η τιμή 2.

Minimax

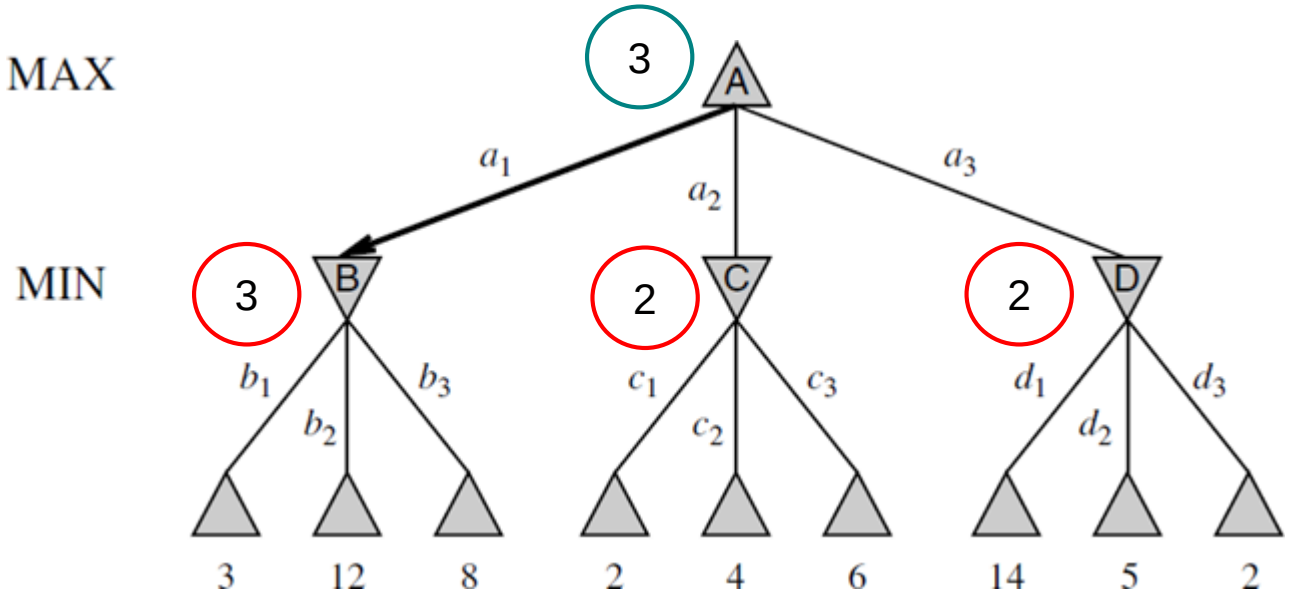
- Παράδειγμα 2 (συν.)



Ο κόμβος D είναι επίσης στην σειρά του MIN. Συνεπώς, επιλέγεται η πιο **μικρή** από τις αξίες των απογόνων του, δηλαδή η τιμή 2.

Minimax

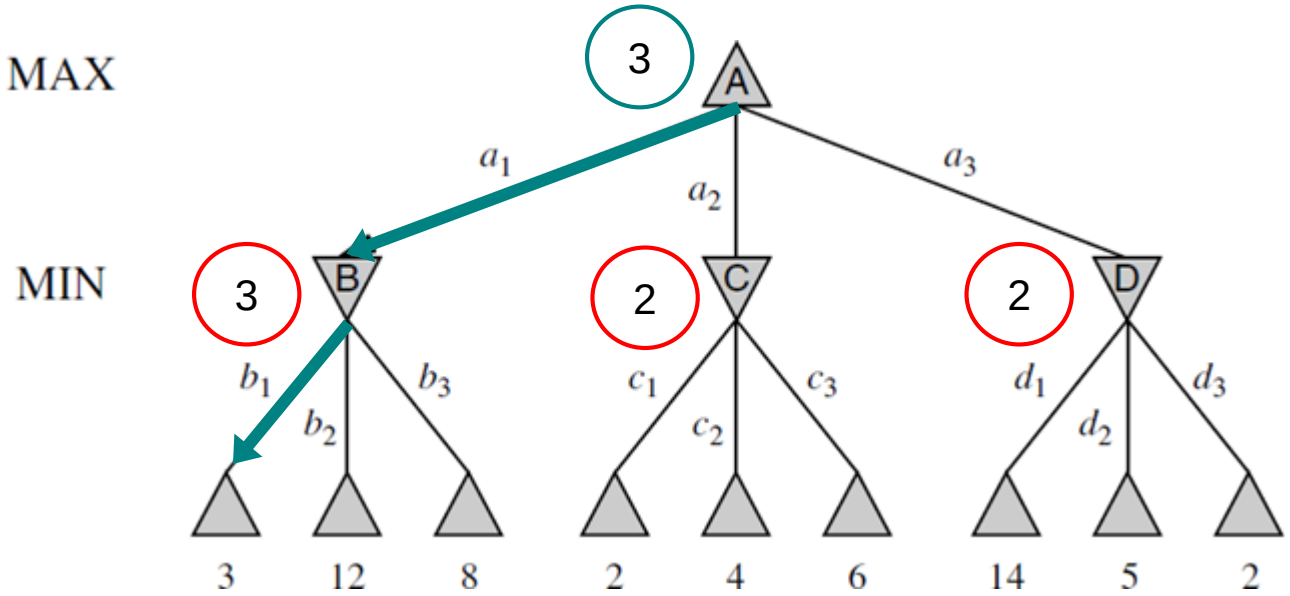
- Παράδειγμα 2 (συν.)



Ο κόμβος A είναι στην σειρά του MAX. Συνεπώς, επιλέγεται η πιο **μεγάλη** από τις αξίες των απογόνων του, δηλαδή η τιμή 3.

Minimax

- Παράδειγμα 2 (συν.)



Συνεπώς το βέλτιστο μονοπάτι είναι το
$a_1 \rightarrow b_1$

Minimax

➢ Χαρακτηριστικά

Ο αλγόριθμος εγγυάται την εύρεση των βέλτιστων κινήσεων ακόμη κι αν ο αντίπαλος κάνει επίσης τις καλύτερες κινήσεις

Βασίζεται στην **εξαντλητική αναζήτηση** όλου του δένδρου παιχνιδιού από την ρίζα ή από τον τρέχοντα κόμβο και κάτω

- Η πλήρης αυτή αναζήτηση συχνά είναι **ανέφικτη** (π.χ. σκάκι)

Η αναζήτηση **πρώτα σε βάθος** (DFS) στον χώρο καταστάσεων από την κατάσταση στην οποία βρισκόμαστε δεν σταματά στην πρώτη τελική κατάσταση που θα βρεθεί αλλά τις διερευνά **όλες**.

Στον κόμβο εκκίνησης επιλέγεται η κίνηση που οδηγεί στο βέλτιστο όφελος.

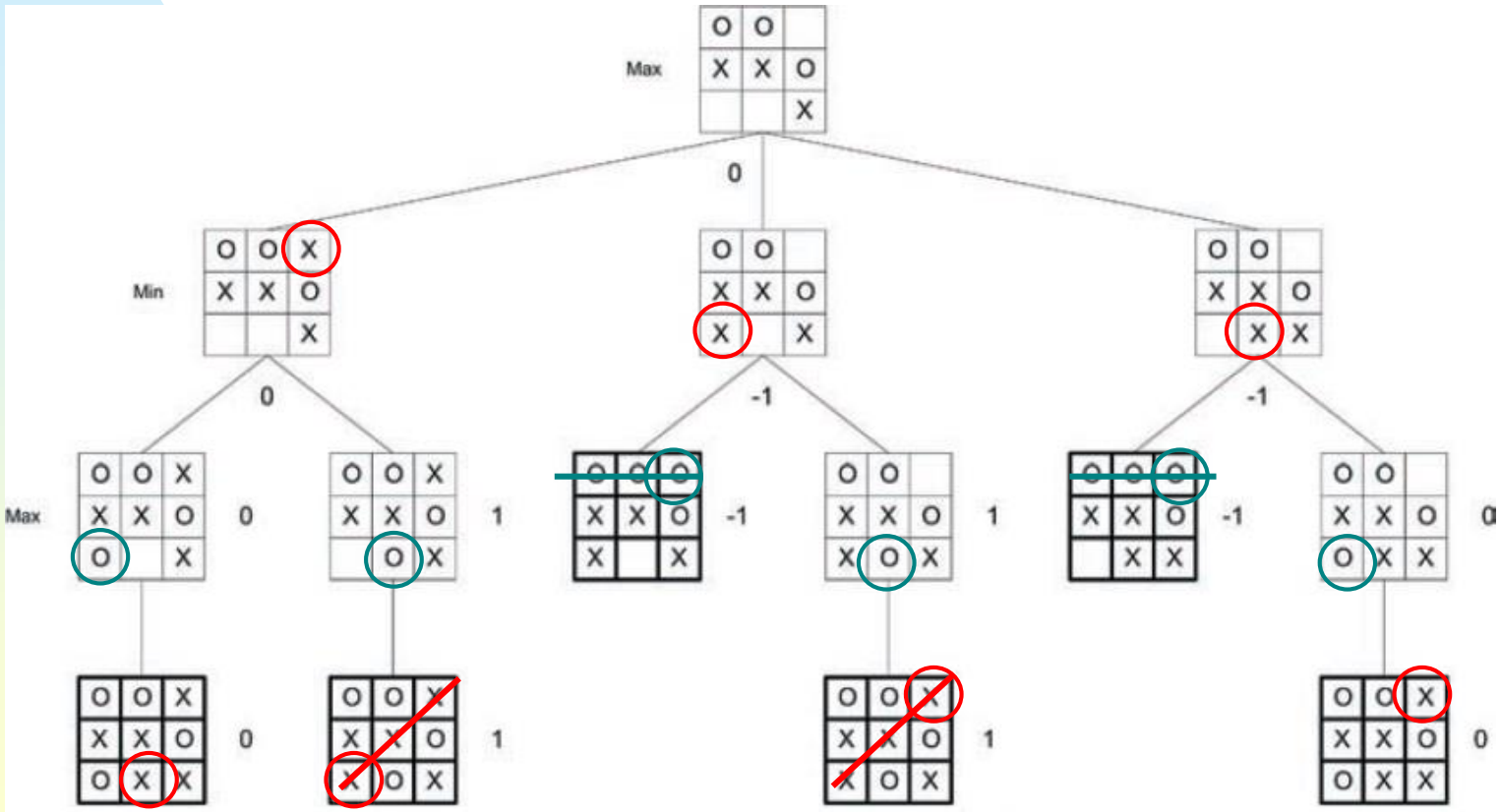
Αν στην συνέχεια ο αντίπαλος **δεν** επιλέξει τις βέλτιστες κινήσεις, τότε το όφελός μας θα είναι ακόμη **μεγαλύτερο**

Η **χρονική πολυπλοκότητα** είναι $\boldsymbol{O(b^m)}$

Η **χωρική πολυπλοκότητα** είναι $\boldsymbol{O(bm)}$

Minimax

- Παράδειγμα 3



Αποκοπή Alpha-Beta

Γενικά

Το πλήθος των καταστάσεων στον Minimax είναι **εκθετικό** ως προς το βάθος του δέντρου αναζήτησης.

Η αποκοπή Alpha-Beta επιτρέπει την μείωση του δέντρου στο **μισό**.

Βασική ιδέα
Μπορούμε να υπολογίσουμε το βέλτιστο μονοπάτι Minimax **χωρίς** να ελέγξουμε **όλο** το δέντρο αναζήτησης

Αποκόπτονται τμήματα του δέντρου τα οποία **δεν επηρεάζουν** τον υπολογισμό της τελικής βέλτιστης διαδρομής.

Αποκοπή Alpha-Beta

Μεθοδολογία

Εντοπισμός κόμβων που **δεν θα επιλεγούν ποτέ** από τον Minimax **οποιαδήποτε τιμή κι αν έχουν οι απόγονοί τους.**

Παράδειγμα

Το $Max\ (3, Min(2, x, y)\ \ldots)$ είναι πάντα $\geq\ 3$

Το $Min\ (2, Max(3, x, y)\ \ldots)$ είναι πάντα ≤ 2

Αυτά **ισχύουν ασχέτως από την τιμή των x και y**.

Αποκοπή Alpha-Beta

- Μεθοδολογία

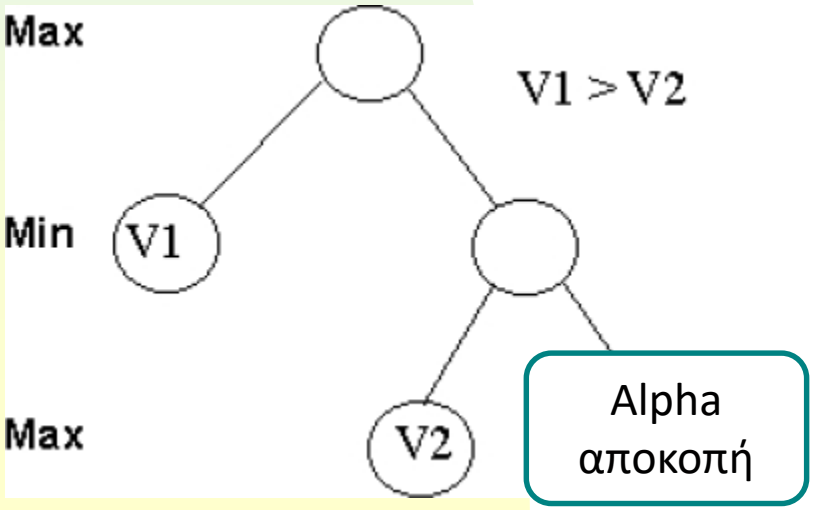
Alpha είναι η υψηλότερη τιμή που έχουμε βρει μέχρι στιγμής για τον κόμβο MAX

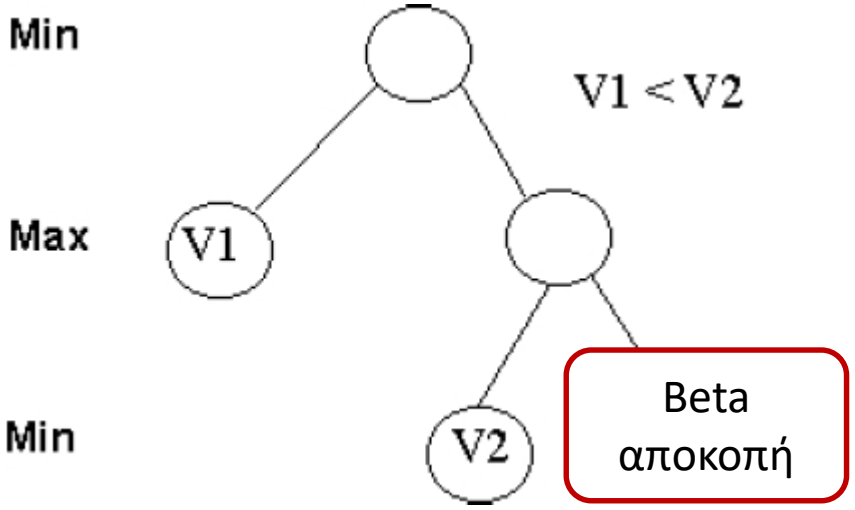
Beta είναι η χαμηλότερη τιμή που έχουμε βρει μέχρι στιγμής για τον κόμβο MIN

Κανόνας

Σε κόμβο MAX αποκόπτονται τιμές **μικρότερες** από $Alpha$

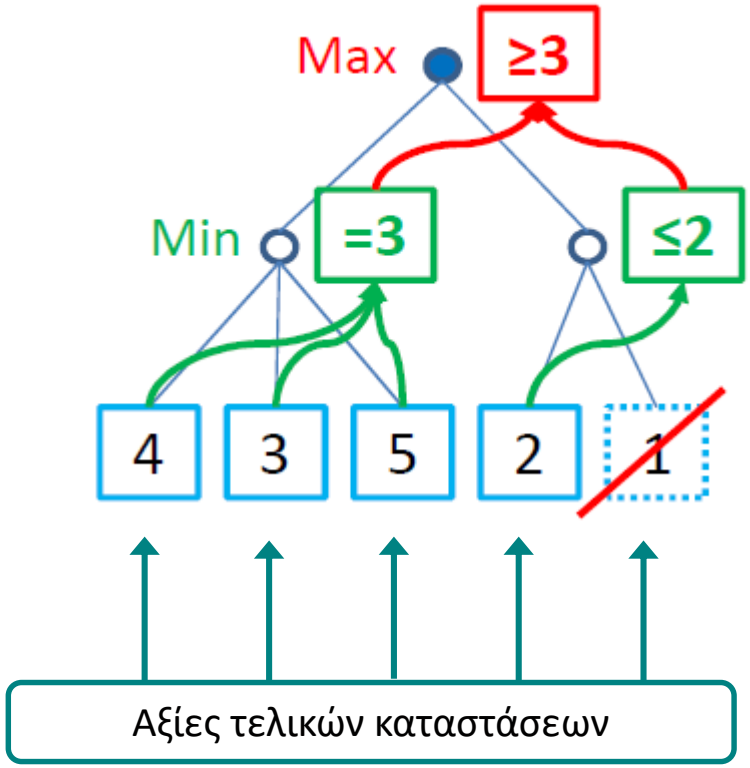
Σε κόμβο MIN αποκόπτονται τιμές **μεγαλύτερες** από $Beta$





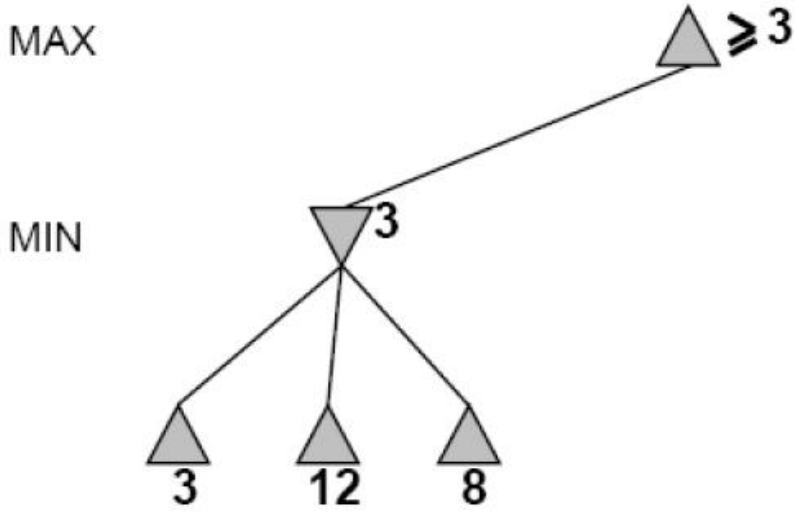
Αποκοπή Alpha-Beta

- Παράδειγμα 1



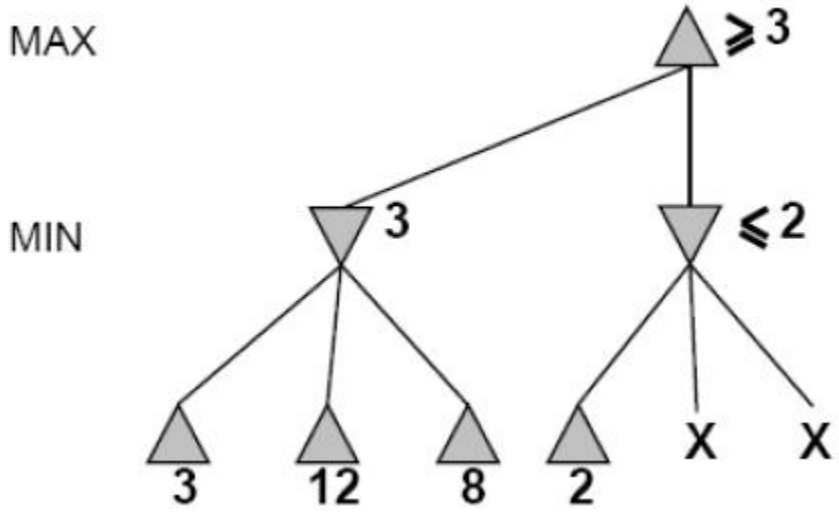
Αποκοπή Alpha-Beta

- Παράδειγμα 2



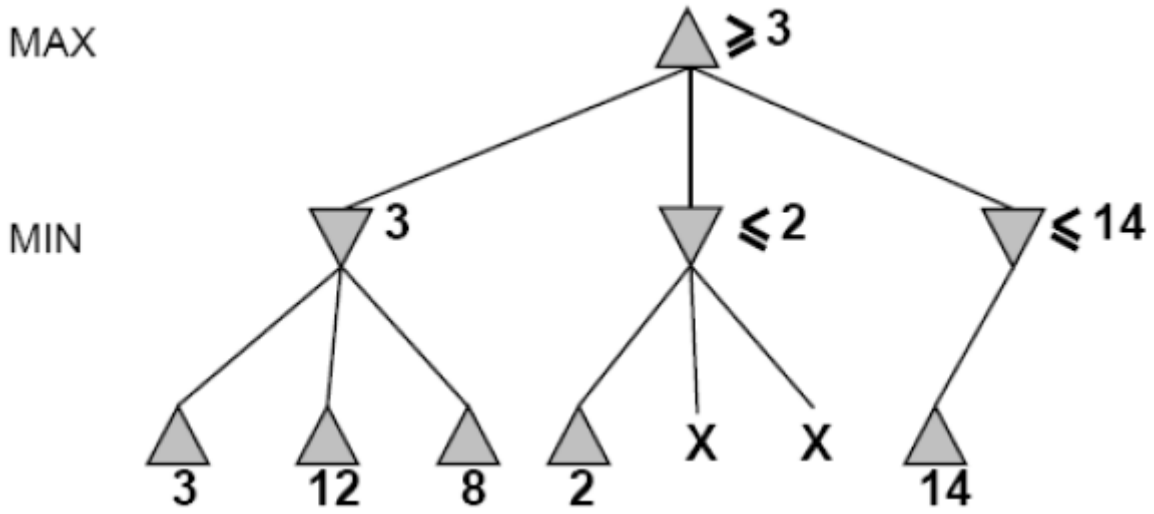
Αποκοπή Alpha-Beta

- Παράδειγμα 2 (συν.)



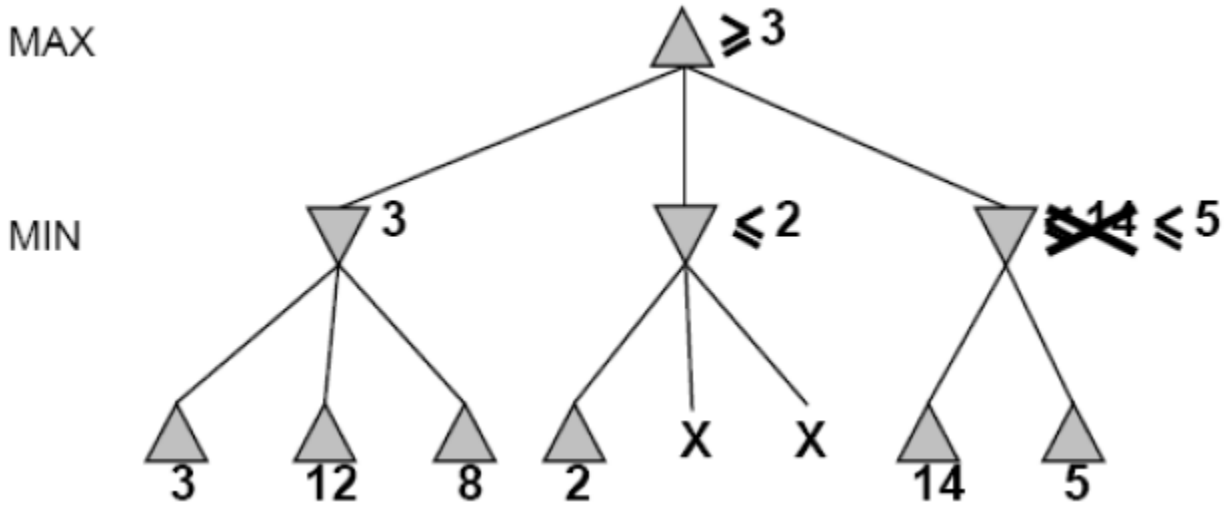
Αποκοπή Alpha-Beta

- Παράδειγμα 2 (συν.)



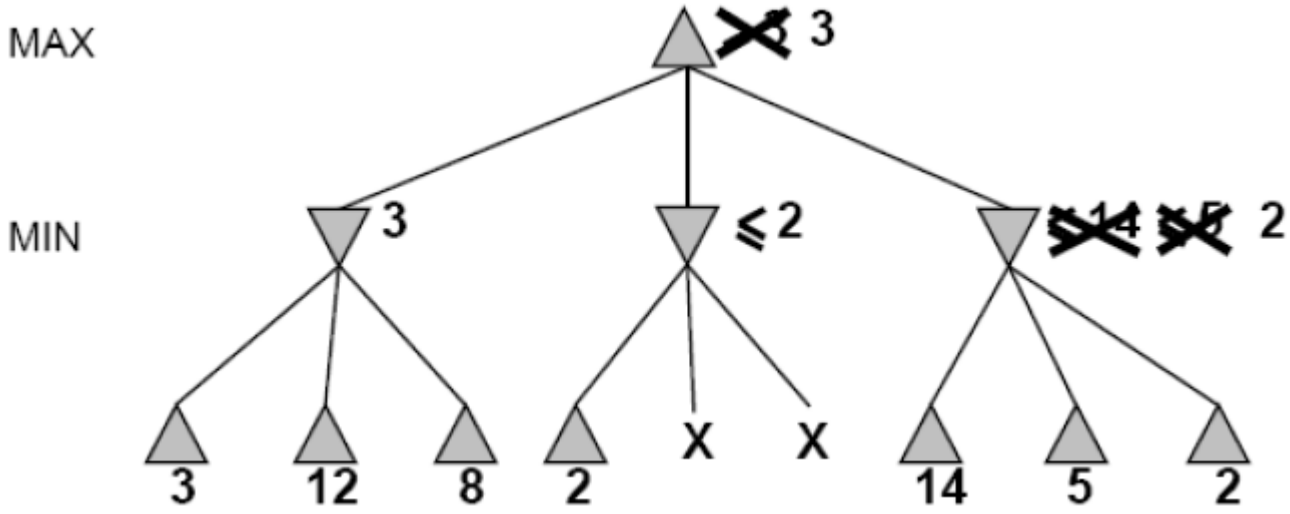
Αποκοπή Alpha-Beta

- Παράδειγμα 2 (συν.)



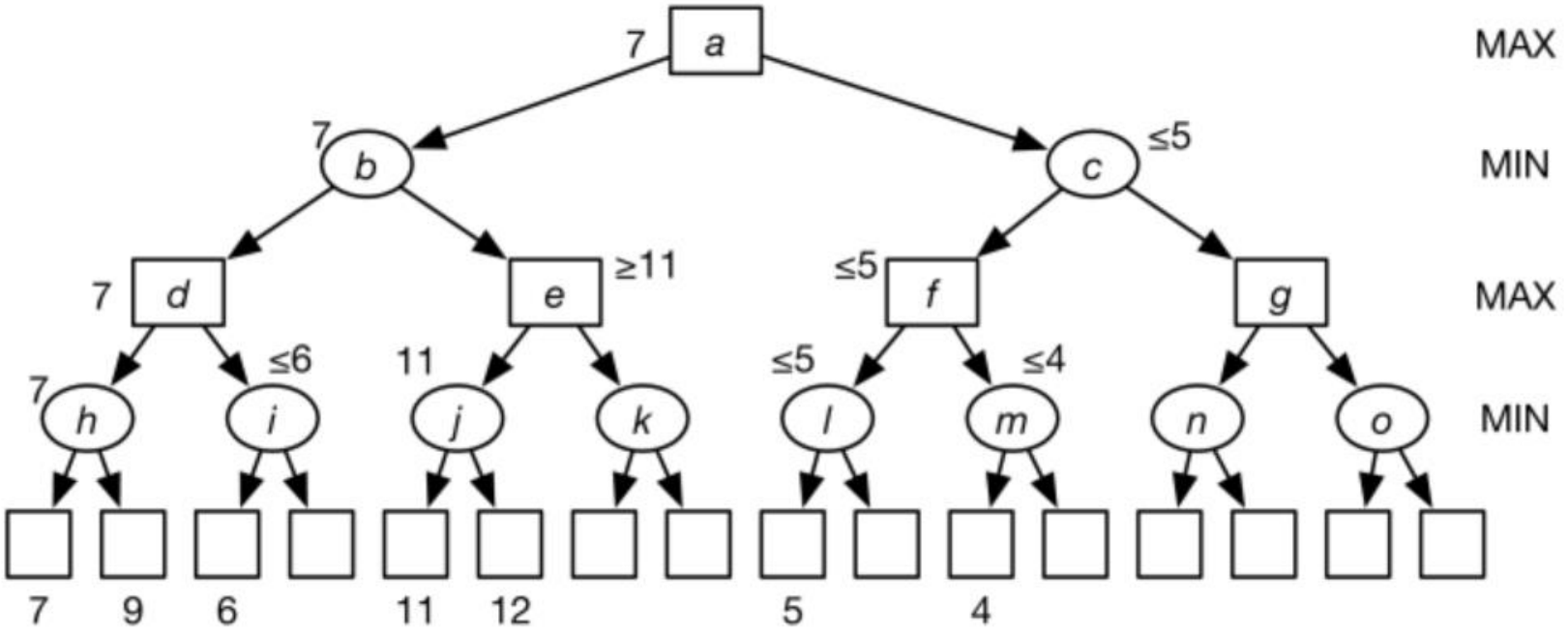
Αποκοπή Alpha-Beta

- Παράδειγμα 2 (συν.)



Αποκοπή Alpha-Beta

- Παράδειγμα 3



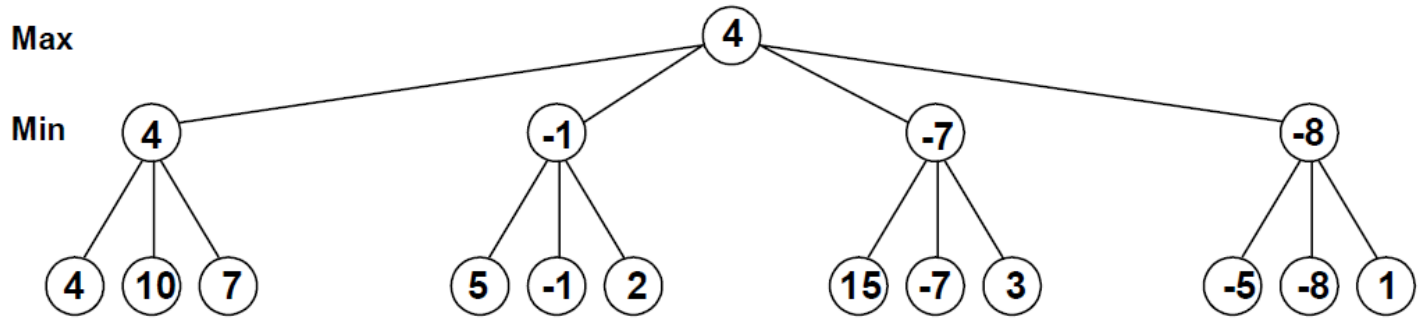
Έστω αναζήτηση αριστερού πρώτου βάθους αυτού του δέντρου.
Η τιμή του κόμβου h είναι 7, επειδή είναι το ελάχιστο των 7 και 9.
Λαμβάνοντας υπόψη το αριστερότερο απόγονο του i με τιμή 6, γνωρίζουμε ότι η τιμή του i είναι μικρότερη ή ίση με 6.
Επομένως, στο κόμβος d, ο παράγοντας μεγιστοποίησης θα πάει αριστερά.
Άρα, δεν χρειάζεται να αξιολογήσουμε το άλλο απόγονο του i.

Αποκοπή Alpha-Beta

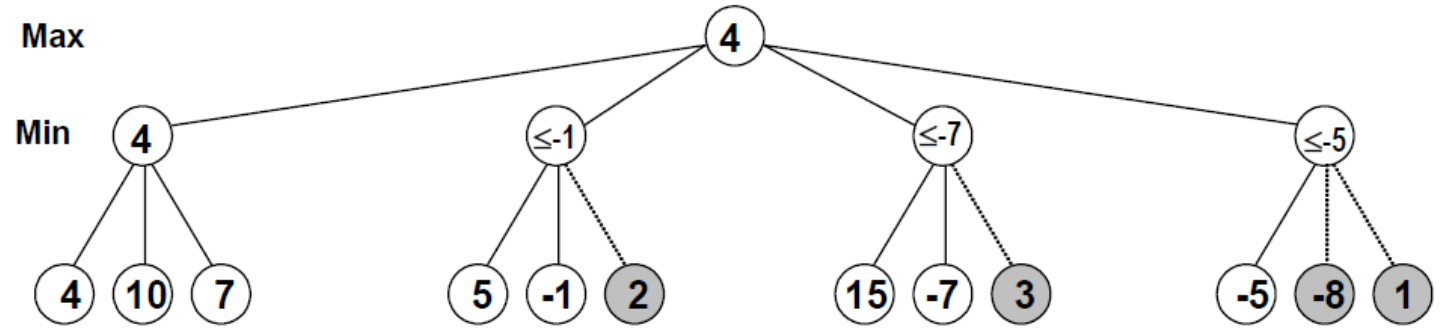
➢ Παράδειγμα 4

Σύγκριση **Minimax** με **Alpha-Beta**

Minimax



Alpha-Beta



Αποκοπή Alpha-Beta

Αποτελεσματικότητα

Είναι εγγυημένο ότι υπολογίζει την ίδια ελάχιστη τιμή για τον ριζικό κόμβο με τον υπολογισμό από το Minimax

Στη **χειρότερη περίπτωση**, η αποκοπή Alpha-Beta δεν κάνει κλάδεμα, εξετάζοντας b^d κόμβους φύλλων, όπου κάθε κόμβος έχει b απόγονους και ο καθένας μπορεί να λάβει d διαφορετικές τιμές

Στην **καλύτερη περίπτωση**, η αποκοπή Alpha-Beta εξετάζει μόνο $(2b)^{d/2}$ κόμβους φύλλων.

- για σταθερό αριθμό των κόμβων φύλλων μπορεί να γίνει αναζήτηση σε **διπλάσιο βάθος** από το Minimax

Η καλύτερη περίπτωση ισχύει όταν ο **καλύτερος** διάδοχος κόμβος είναι και ο **πρώτος** (αριστερά)

- στους κόμβους MAX δημιουργείται πρώτα ο απόγονος με τη μεγαλύτερη τιμή
- στους κόμβους MIN δημιουργείται πρώτα ο απόγονος με τη μικρότερη τιμή

Συναρτήσεις αξιολόγησης

➢ Αποκοπή (cut-off)

Πρόβλημα με την κλίμακα του δέντρου αναζήτησης

- Στα περισσότερα παίγνια το δέντρο αναζήτησης του παιχνιδιού αυξάνεται με **εκθετικό** τρόπο.
- Ακόμη και τεχνικές όπως οι Minimax και Alpha-Beta μπορούν να διαχειριστούν βάθη δέντρου μεταξύ 5 και 10 (ανάλογα με την φύση του παιχνιδιού).
- Οπότε μπορούν να είναι χρήσιμες μόνο εάν η τρέχουσα κατάσταση είναι **κοντά στο τέλος** του παιχνιδιού.

Αποκόπτεται (**cutting-off**) το δένδρο αναζήτησης σε ένα συγκεκριμένο βάθος

Οι κόμβοι σε αυτό το βάθος θεωρούνται **τερματικοί κόμβοι** με τιμή που καθορίζεται από μια **ευριστική συνάρτηση αξιολόγησης**

Συναρτήσεις αξιολόγησης

➢ Ευριστικές συναρτήσεις αξιολόγησης

Μια **ευριστική συνάρτηση αξιολόγηση** (**heuristic evaluation function**) $h(n)$ προσδιορίζει την τιμή αξίας σε **οποιοδήποτε κόμβο** n του δένδρου αναζήτησης.

Παράδειγμα: Σκάκι

- Αξία πιονιών
  π.χ. βασιλιάς: 10, άλογο: 5, πιόνι: 1
- Θέσεις κομματιών
  π.χ. επιπλέον 2 πόντοι για κάθε πιόνι στα 4 κεντρικά τετράγωνα
- Απειλές
  π.χ. 3 επιπλέον πόντοι για κάθε απειλή πιονιού αντιπάλου
  20 επιπλέον πόντοι για απειλή προς βασιλιά

Συνήθως χρησιμοποιείται **συνδυασμός** k ευριστικών συναρτήσεων

$$f(n) = w_1 h_1(n) + w_2 h_2(n) + \cdots + w_k h_k(n)$$

- Προσδιορισμός **βαρών** w_i π.χ. με χρήση τεχνικών μηχανικής μάθησης μέσω πολλών **παραδειγμάτων εκπαίδευσης**

Συναρτήσεις αξιολόγησης

Ευριστικές συναρτήσεις αξιολόγησης (συν.)

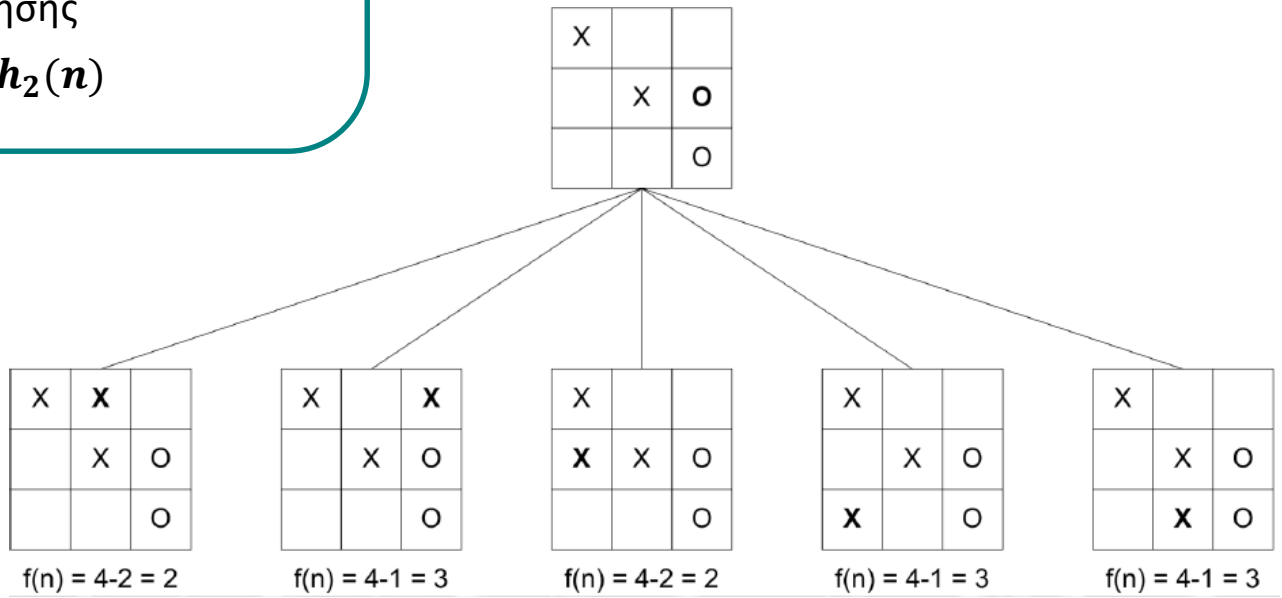
Παράδειγμα: Τρίλιζα

$\boldsymbol{h_1(n)}$=πλήθος από πιθανές νίκες (οριζόντιες, κάθετες , διαγώνιες) για τον παίκτη MAX (X)

$\boldsymbol{h_2(n)}$=πλήθος από πιθανές νίκες (οριζόντιες, κάθετες , διαγώνιες) για τον παίκτη MIN (O)

Ευριστική συνάρτηση αξιολόγησης

$$\boldsymbol{f(n) = h_1(n) - h_2(n)}$$



Συναρτήσεις αξιολόγησης

Ευριστικές συναρτήσεις αξιολόγησης (συν.)

Παράδειγμα: Τρίλιζα

Εναλλακτικός υπολογισμός

$\boldsymbol{h_{x1}(n)}$=αριθμός γραμμών, στηλών ή διαγωνίων με ένα X και κανένα O

$\boldsymbol{h_{x2}(n)}$=αριθμός γραμμών, στηλών ή διαγωνίων με δύο X και κανένα O

$\boldsymbol{h_{y1}(n)}$=αριθμός γραμμών, στηλών ή διαγωνίων με ένα O και κανένα X

$\boldsymbol{h_{y2}(n)}$=αριθμός γραμμών, στηλών ή διαγωνίων με δύο O και κανένα X

Ευριστική συνάρτηση αξιολόγησης

$$\boldsymbol{f(n) = \left(3h_{x2}(n) + h_{x1}(n)\right) - \left(3h_{y2}(n) + h_{y1}(n)\right)}$$

X	X	
	X	O
		O

$$f(n) = (3 \cdot 2 + 2) - (3 \cdot 1 + 1) = 8 - 4 = 4$$

Συναρτήσεις αξιολόγησης

➢ Ευριστικές συναρτήσεις αξιολόγησης (συν.)

Υπολογισμός Minimax με cut-off

- Εάν ο κόμβος n είναι τερματικός κόμβος τότε $f(n)$ είναι η τιμή αξίας του κόμβου
- Εάν ο κόμβος n είναι στην σειρά του MAX τότε $f(n)$ είναι η μέγιστη τιμή των ευριστικών τιμών των απογόνων του n
- Εάν ο κόμβος n είναι στην σειρά του MIN τότε $f(n)$ είναι η ελάχιστη τιμή των ευριστικών τιμών των απογόνων του n

Μια ευριστική συνάρτηση αποτελεί **εκτίμηση** της αξίας με την προϋπόθεση ότι οι δύο αντίπαλοι έχουν πραγματοποιήσει μέχρι στιγμής τις κινήσεις με βάση τον αλγόριθμο Minimax

Στην πράξη, ακόμα κι αν χρησιμοποιηθεί η ίδια ευριστική, ο αντίπαλος θα ψάξει όταν έρθει η σειρά του **μέχρι ένα επίπεδο πιο κάτω**

(συνεπώς θα αποφασίσει βάσει **άλλου δέντρου** οπότε μπορεί η κίνηση να είναι αλλιώτικη από αυτήν που αναμενόταν)

Παιχνίδια με παράγοντα τύχης

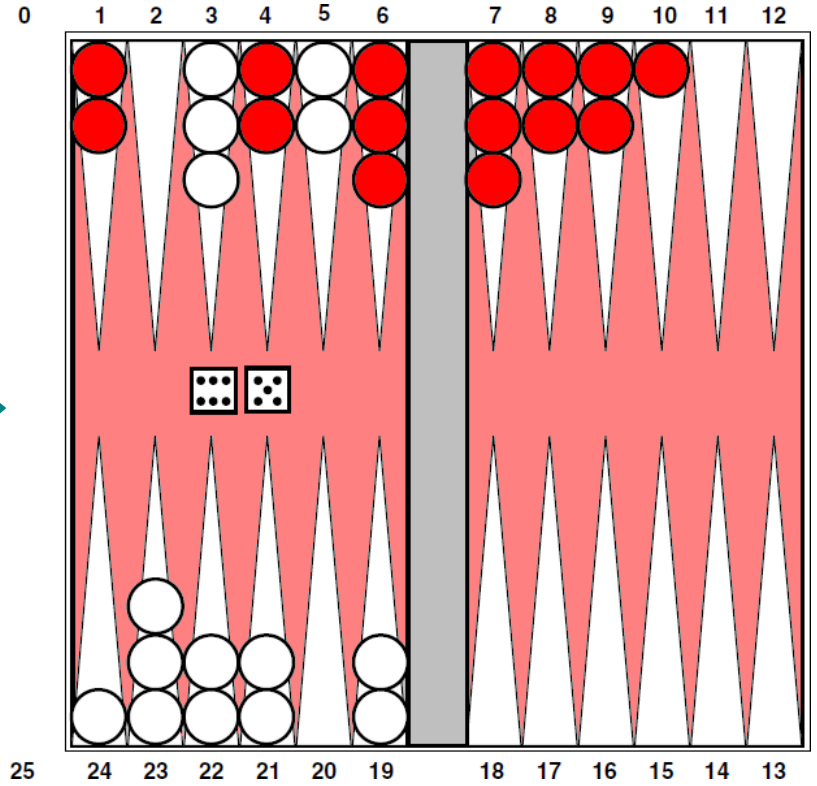
Στοχαστικά παιχνίδια

Σε πολλά παιχνίδια η εξέλιξή τους εξαρτάται και από τον παράγοντα τύχη

Παράδειγμα: τάβλι

Οι επόμενες επιτρεπτές κινήσεις κάθε παίκτη καθορίζονται από την ρίψη ζαριών

Για ζάρια
6 και 5
Οι πιθανές κινήσεις είναι
(5–10, 5–11)
(5–11, 19–24)
(5–10, 10–16)
(5–11, 11–16)



Οι κινήσεις του λευκού είναι γνωστές

Οι κινήσεις του κόκκινου εξαρτώνται από την ρίψη του δικού του ζαριού

Παιχνίδια με παράγοντα τύχης

- Στοχαστικά παιχνίδια (συν.)

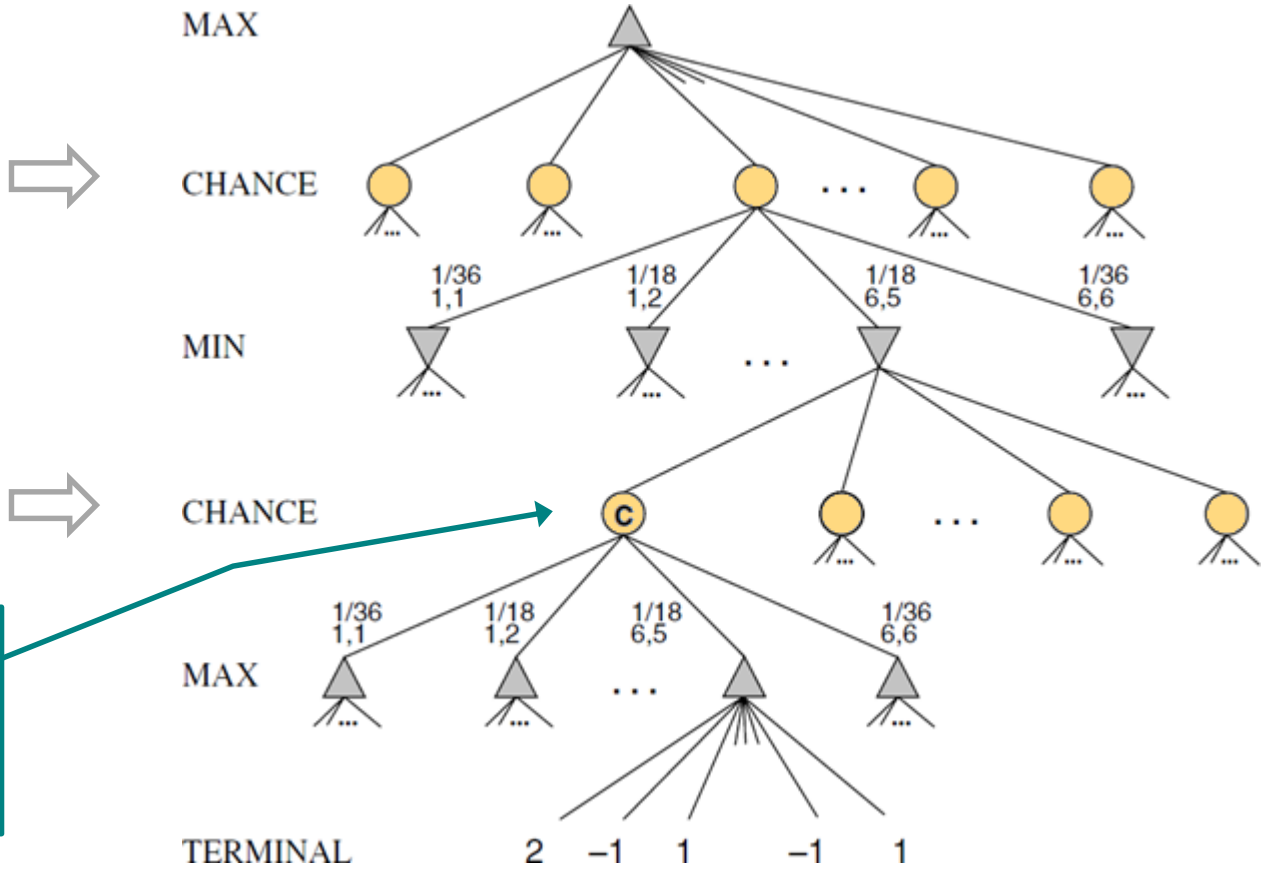
Προσέγγιση: expectiminimax

Χρήση ενός **επιπλέον επιπέδου** ανάμεσα στα MIN και MAX που περιέχει τις πιθανές τυχαίες τιμές.

Πολυπλοκότητα

$O(b^m r^m)$

όπου r το πλήθος των τυχαίων καταστάσεων



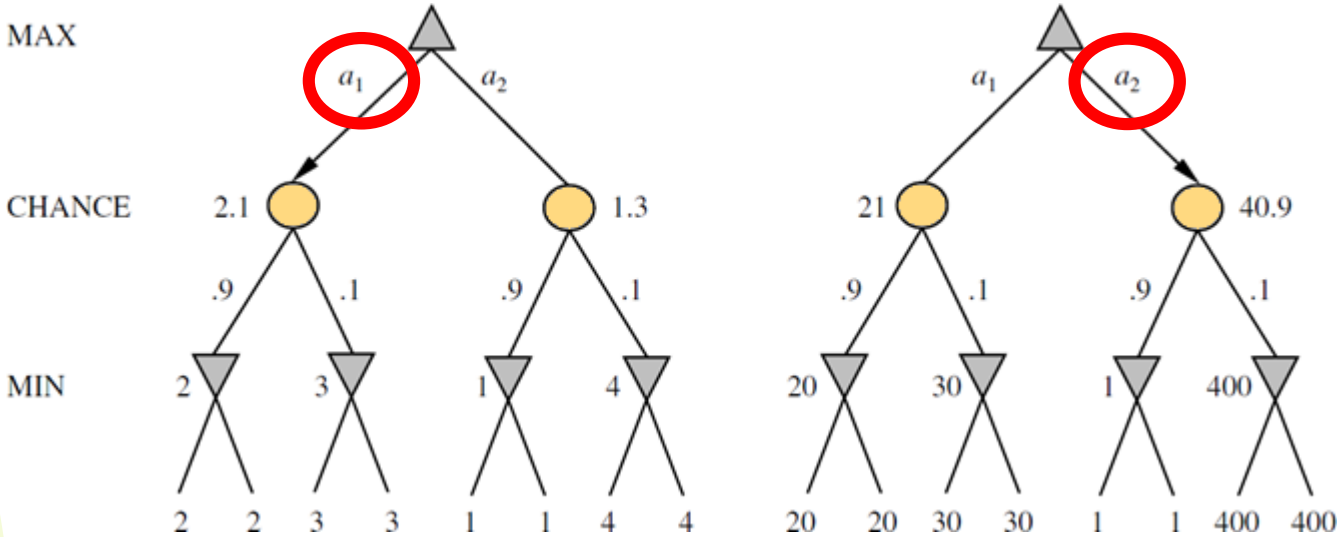
Σταθμισμένο (με τις επιμέρους πιθανότητες) άθροισμα των απογόνων κόμβων

Παιχνίδια με παράγοντα τύχης

- Στοχαστικά παιχνίδια (συν.)

Απαιτείται προσοχή στην χρήση **ευριστικών συναρτήσεων** για την απόδοση μεγαλύτερων τιμών στις καλύτερες καταστάσεις

Παράδειγμα



Οι υπερβολικά μεγάλες τιμές της ευριστικής συνάρτησης μεταβάλουν το αποτέλεσμα από a_1 και a_2

Αναζήτηση με αντιπαλότητα

Σύνοψη

Σε παιχνίδια **μηδενικού αθροίσματος** δύο παικτών με πλήρη εικόνα των καταστάσεων, ο αλγόριθμος **Minimax** δίνει τις **βέλτιστες** κινήσεις μέσω μιας αναζήτησης πρώτα σε βάθος στο δέντρο αναζήτησης του παιχνιδιού.

Ο αλγόριθμος **Alpha-Beta** υπολογίζει την ίδια βέλτιστη λύση αλλά με πολύ μεγαλύτερη αποτελεσματικότητα μέσω της αποκοπής τμημάτων του δένδρου αναζήτησης που είναι προφανώς περιττά.

Συνήθως, δεν είναι εφικτή η εξέταση ολόκληρου του δέντρου (ακόμη και με αποκοπή Alpha-Beta) οπότε διακόπτεται η αναζήτηση σε κάποιο βάθος του δένδρου και εφαρμόζεται κάποια **ευριστική συνάρτηση αξιολόγησης** που εκτιμά την κατάσταση.

Παιχνίδια που εμπεριέχουν και τον παράγοντα της τύχης μπορούν να αντιμετωπιστούν με τον αλγόριθμο **expecti-minimax** που αξιολογεί κάθε κόμβο υπολογίζοντας το άθροισμα όλων των απογόνων του, **σταθμισμένο** με την πιθανότητα του κάθε απογόνου.

Αναζήτηση με αντιπαλότητα

Deep Blue

(από Wikipedia)

Δημιουργήθηκε το 1997

Hardware

- 30 IBM RS/6000 επεξεργαστές
- 480 ειδικοί επεξεργαστές για **σκάκι**
- Παράλληλη επεξεργασία
- Αναζήτηση εις βάθος στο δέντρο
- Δημιουργία και ταξινόμηση πιθανών κινήσεων
- Εκτίμηση θέσεων

Αλγόριθμος

- Επαναληπτική αναζήτηση εμβάθυνσης με αναζήτηση alpha-beta
- Χρήση βάσεων δεδομένων με 4000 ανοίγματα και 700000 παιχνίδια γκραντ-μάστερ
- Βάση με όλες τις περιπτώσεις με πέντε τελευταία κομμάτια και πολλές περιπτώσεις με έξι κομμάτια

Αναζήτηση με αντιπαλότητα

Deep Blue (συν.)

Ιστορικό

- Στις 11 Μαΐου **1997**, ο Deep Blue νίκησε στον έκτο γύρο μπαράζ τον Κασπάροφ, καθώς στους πέντε αγώνες ήταν ισόπαλοι με 2½-2½. Έτσι το τελικό σκορ ήταν **3½–2½**.
- Ο Deep Blue έγινε η πρώτη **σκακιστική μηχανή** που νίκησε τον παγκόσμιο πρωταθλητή σκακιού σε ένα τουρνουά υπό κανονικές συνθήκες διεξαγωγής αγώνων και κανονικά χρονικά όρια για κάθε παίχτη.
- Μετά την ήττα του, ο Κασπάροφ είπε ότι **διέκρινε βαθιά νοημοσύνη** και δημιουργικότητα στις κινήσεις της μηχανής, και μάλιστα ισχυρίστηκε ότι στον δεύτερο αγώνα, πίσω από τις κινήσεις του υπολογιστή κρύβονταν **άνθρωποι σκακιστές**, κάτι που ήταν ενάντια στους κανόνες του αγώνα. Η IBM ωστόσο αρνήθηκε ότι ο Deep Blue έκλεβε, λέγοντας ότι η μόνη ανθρώπινη παρέμβαση στη μηχανή γινόταν στο μεσοδιάστημα των αγώνων.
- Ο Κασπάροφ **απαίτησε αντίτυπα των αρχείων καταγραφής** του Deep Blue, όμως η IBM τότε αρνήθηκε, αν και αργότερα δημοσίευσε στο ίντερνετ αυτά τα αρχεία.
- Ο Κασπάροφ ζήτησε να γίνει **και τρίτη σειρά αγώνων**, αλλά η IBM **αρνήθηκε** και πάλι, **αποσυναρμολογώντας** τον Deep Blue.