



## ΕΝΟΤΗΤΑ 1.6

### Αλγόριθμοι Εύρεσης Λύσης σε Παίγνια Δυο Αντιπάλων

Διδάσκων Ενότητας:  
Κώστας Κοντογιάννης

ΜΕΡΟΣ 1: Επίλυση Προβλημάτων στο Χώρο των καταστάσεων

## Περίληψη Ενότητας 1.6



2

### Θέματα

- Αλγόριθμοι Εύρεσης Λύσης σε Παίγνια Δύο Αντιπάλων

### Υλικό

- Διαφάνειες παρουσίασης
- Κεφάλαιο 5 Βιβλίου

## Βιβλιογραφία - Αναφορές



3

- Υλικό για αυτή τη διάλεξη συγκεντρώθηκε
  - ▣ από το βιβλίο «Τεχνητή Νοημοσύνη», Ι. Βλαχάβας κ.α, Εκδόσεις Γκιούρδας

## Εισαγωγικά (i)



4

- Σε πολλά προβλήματα της ΤΝ, η εξέλιξη των καταστάσεων εξαρτάται από δύο δια-φορετικά σύνολα τελεστών μετάβασης που εφαρμόζονται εναλλάξ από δύο ενεργά συστήματα (πράκτορες-agents)
- Αυτά τα προβλήματα αναφέρονται και ως ανταγωνιστικά παίγνια ή παίγνια δύο αντιπάλων (*adversary ή two-person games*)
  - ▣ Επιλύονται με ειδικούς αλγορίθμους αναζήτησης, βασικότεροι των οποίων είναι ο minimax και ο alpha-beta
- Ο όρος "παίγνιο" αφορά την περιγραφή του τρόπου με τον οποίο παίζεται το παιχνίδι και περιλαμβάνει:
  - ▣ τα αντικείμενα που υπάρχουν (για παράδειγμα, τα πούλια, το ταμπλώ, κτλ.) καθώς και
  - ▣ το σύνολο των κανόνων (για παράδειγμα, των επιτρεπτών κινήσεων, των συνθηκών νίκης, κτλ.) που το διέπουν.
- Αντίθετα, με τον όρο "παιχνίδι" χαρακτηρίζεται μία συγκεκριμένη παρτίδα του παιχνιδιού.
  - ▣ Γνωστά παραδείγματα παιγνίων είναι το σκάκι, η τρίλιζα, το τάβλι, κτλ.

## Εισαγωγικά (ii)



5

- Σε ένα παίγνιο δύο ατόμων το πρόβλημα ορίζεται ως εξής:
  - Μια κατάσταση παριστάνει τη διάταξη των πιονιών σε κάποια χρονική στιγμή.
  - Ο χώρος καταστάσεων αποτελείται από όλες αυτές τις πιθανές επιτρεπτές καταστάσεις.
  - Οι τελεστές μετάβασης είναι οι επιτρεπτές κινήσεις που καθορίζονται από τους κανόνες του παιχνιδιού. Οι κανόνες του παιχνιδιού παίζουν και το ρόλο των προϋποθέσεων αυτών των τελεστών.
  - Οι τελικές καταστάσεις του παιχνιδιού έχουν γνωστά χαρακτηριστικά (π.χ. ματ στο σκάκι).
- Έστω ότι κάποιος συμμετέχει σε ένα παιχνίδι και είναι η σειρά του να κάνει μία κίνηση.
  - Αν θέλει να κερδίσει, θα επιδιώξει να κάνει την καλύτερη κίνηση για αυτόν, η οποία αντίστοιχα θα είναι και η χειρότερη για τον αντίπαλο.
  - Ο τρόπος που πιθανά σκέφτεται κάποιος για να παίξει είναι ο εξής: "Αν κάνω αυτή την κίνηση, τότε ο αντίπαλος θα κάνει εκείνη, αν όμως κάνω την άλλη κίνηση, τότε θα κάνει αυτήν, ..." κ.ο.κ.
  - Ο συλλογισμός αυτός αντιστοιχεί στη δημιουργία ενός δένδρου με όλες τις εναλλακτικές κινήσεις που πηγάζουν από την παρούσα κατάσταση.
  - Το δένδρο αυτό ονομάζεται **δένδρο του παιχνιδιού (game tree)**.

## Εισαγωγικά (iii)



6

- Το χαρακτηριστικό του είναι ότι οι κινήσεις δύο διαδοχικών επιπέδων ανήκουν σε διαφορετικό παίκτη, γιατί οι παίκτες παίζουν εναλλάξ.
  - Το δένδρο αυτό αναπτύσσεται νοερά από τον παίκτη του οποίου είναι η σειρά να παίξει.
  - Αυτή η διαδικασία μπορεί να συνεχιστεί μέχρι κάποιο βάθος, συνήθως μικρό, και να σταματήσει εκεί όπου αρχίζει η αξιολόγηση των καταστάσεων με βάση κάποια κριτήρια.
  - Ο παίκτης αξιολογεί τις καταστάσεις που θα προκύψουν από πιθανές διαφορετικές εξελίξεις του παιχνιδιού και αποφασίζει ποια από τις εναλλακτικές κινήσεις θα τον οδηγήσει σταδιακά στην πιο ευνοϊκή για αυτόν εξέλιξη.
- Έστω ότι ένας από τους δύο αντιπάλους είναι ο υπολογιστής.
  - Το πρόγραμμα αναζήτησης και ο άνθρωπος πρέπει να εξετάσουν και να θυμούνται όλες τις πιθανές κινήσεις που παράγονται από μία (αρχική) κατάσταση.
  - Ένα τέτοιο πρόγραμμα μπορεί να προβλέψει την έκβαση του παιχνιδιού μετά από 10 κινήσεις (βάθος 10 του δένδρου παιχνιδιού), ενώ ο άνθρωπος σταματά συνήθως μετά την πρόβλεψη 2 ή 3 κινήσεων.

## Εισαγωγικά (iv)



7

- Παραδόξως, οι διαφορές αυτές δεν κάνουν κάποιο πρόγραμμα κυρίαρχο οποιουδήποτε παιχνιδιού.
  - Π.χ., μόλις πρόσφατα κατασκευάστηκε το καλύτερο σύστημα, το Deep Blue (συνδυασμός λογισμικού και υλικού) που μπορεί, κάποιες φορές, να νικήσει τον παγκόσμιο πρωταθλητή στο σκάκι.
  - Ο λόγος για την ανωτερότητα των ανθρώπων στο σκάκι είναι από τη μία μεριά απλός να περιγραφεί, αλλά από την άλλη πολύ δύσκολος να υλοποιηθεί σαν πρόγραμμα.
- Η ανωτερότητα τους έγκειται στους εξής παράγοντες:
  - Οι μαίτρ του σκάκι έχουν τη δυνατότητα να επιλέγουν με κάποιον ευριστικό τρόπο τις εναλλακτικές κινήσεις, αλλά δε μπορούν να μετατρέψουν αυτόν τον τρόπο σε αντίστοιχη ευριστική τιμή.
- Οι άνθρωποι διαθέτουν κάποια διαίσθηση για την κατάληξη του παιχνιδιού αν επιλεγεί κάποια από τις εναλλακτικές κινήσεις η οποία επίσης δε μπορεί να αποδοθεί αριθμητικά ώστε να υπάρξει ευριστικό κλάδεμα του χώρου αναζήτησης.
- Τέλος, με την εμπειρία που αποκτούν, μπορούν να σκέφτονται πολλές φορές εντελώς μηχανικά, ιδίως στα πρώτα και τελευταία στάδια του παιχνιδιού.

## Εισαγωγικά (v)



8

- Το κύριο μέλημα των προγραμμάτων σκάκι είναι να αναζητήσουν το δένδρο του παιχνιδιού σε όσο το δυνατόν μεγαλύτερο βάθος.
  - Αν υπήρχε η δυνατότητα να φτάσουν μέχρι τις τελικές καταστάσεις του παιχνιδιού τότε τα προγράμματα θα ήταν ανίκητα.
  - Ο λόγος που κάτι τέτοιο είναι αδύνατο πηγάζει από τους χρονικούς περιορισμούς μιας παρτίδας και από το πρόβλημα της συνδυαστικής έκρηξης.
- Αν υποθέσουμε ότι από κάθε κατάσταση στο σκάκι παράγονται άλλες 10 καταστάσεις (Μ.Ο.), και ότι μία παρτίδα τελειώνει σε 40 κινήσεις (20 για κάθε παίκτη), τότε οι πιθανές τελικές καταστάσεις θα ήταν της τάξης του 1040.
  - Αν ένας υπέρ-υπολογιστής ξοδεύει 1 nanosecond ( $=10^{-9}$  sec) για να εξετάσει κάθε μία από αυτές, τότε χρειάζεται  $1040 \times 10^{-9} = 1031$  sec συνολικά για να τις εξετάσει όλες.
  - Έχοντας υπόψη πως ένα έτος έχει 3107 sec περίπου, ο υπέρ-υπολογιστής αυτός θα χρειαζόταν 31024 χρόνια για να διαλέξει την επόμενη κίνησή του.
  - Αυτό είναι περίπου και το 1/4 της ηλικίας του σύμπαντος!
- Αφού λοιπόν δε μπορούμε να περιμένουμε τόσο πολύ, οι αλγόριθμοι αναζήτησης που υπάρχουν εξετάζουν το δένδρο του παιχνιδιού μέχρι ένα ορισμένο βάθος, τόσο όσο επιτρέπει ο χρόνος για τη κίνηση.

## Ο Αλγόριθμος Min-Max



9

- Δεδομένης μίας κατάστασης του παιχνιδιού, ο αλγόριθμος αναζήτησης μεγίστου-ελαχίστου (*Minimax*) καλείται να αποφασίσει ποια θα είναι η επόμενη κίνησή του έναντι του αντιπάλου.
- Η εξαντλητική αναζήτηση των δένδρων αναζήτησης είναι ανέφικτη.
- Το ζητούμενο είναι:
  - ▣ Να χτιστεί το δένδρο μέχρι κάποιο βάθος
  - ▣ Να βρεθεί η καλύτερη κίνηση από την παρούσα κατάσταση.
- Το μέτρο της υπεροχής του ενός ή του άλλου αντιπάλου δίνεται από μία ευριστική συνάρτηση που καλείται συνάρτηση αξιολόγησης (*evaluation function*) και η οποία εφαρμόζεται στα φύλλα του δένδρου του παιχνιδιού.
- Ο ένας παίκτης (πρόγραμμα) ονομάζεται *max* και ο άλλος (άλλο πρόγραμμα ή άνθρωπος) ονομάζεται *min*.
- Οι κόμβοι του δένδρου διακρίνονται σε:
  - ▣ Αυτούς που σειρά έχει να παίξει ο *max*
  - ▣ Αυτούς που σειρά έχει να παίξει ο *min*

## Ο Αλγόριθμος Min-Max (Ψευδοκώδικας)



10

**Βήμα 1.** Εφάρμοσε τη συνάρτηση αξιολόγησης σε όλους τους κόμβους-φύλλα του δένδρου.

**Βήμα 2.** Έως ότου η ρίζα του δένδρου αποκτήσει τιμή, επανέλαβε:

**Βήμα 3.** Αρχίζοντας από τα φύλλα του δένδρου και προχωρώντας προς τη ρίζα, μετέφερε τις τιμές προς τους ενδιαμέσους κόμβους του δένδρου ως εξής:

**Βήμα 3.1** Η τιμή κάθε κόμβου *Max* είναι η μέγιστη (*maximum*) των τιμών των κόμβων-παιδιών του.

**Βήμα 3.2** Η τιμή κάθε κόμβου *Min* είναι η ελάχιστη (*minimum*) των τιμών των κόμβων-παιδιών του.

**Βήμα 4.** Καλύτερη κίνηση είναι η κίνηση που οδηγεί στον κόμβο που έδωσε την πιο συμφέρουσα στη ρίζα τιμή (μέγιστη για το *Max*, ελάχιστη για το *Min*).

Ο αλγόριθμος εγγυάται την πιο συμφέρουσα εξέλιξη μετά από κάποιες κινήσεις, έστω και αν ο αντίπαλος διαλέγει τις καλύτερες για αυτόν κινήσεις.

## Ο Αλγόριθμος Min-Max (Σχόλια)



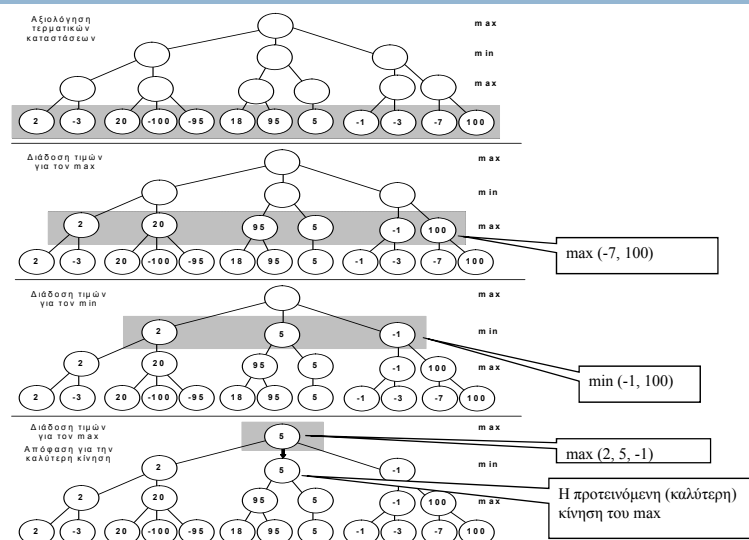
11

- Κατά σύμβαση, ο παίκτης που βρίσκεται στη ρίζα θεωρείται πως είναι ο Max.
- Οι καταστάσεις-φύλλα του δένδρου ονομάζονται και τερματικές καταστάσεις, οι οποίες όμως δεν είναι απαραίτητα τελικές καταστάσεις, απλά αποτελούν το όριο της αναζήτησης.
- Οι τιμές των τερματικών καταστάσεων υπολογίζονται από τη συνάρτηση αξιολόγησης ενώ οι άλλες προκύπτουν από τη διάδοση αυτών.

## Παράδειγμα Min-Max

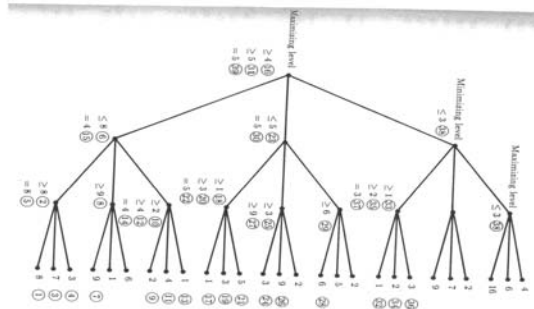


12



## Παράδειγμα Min-Max

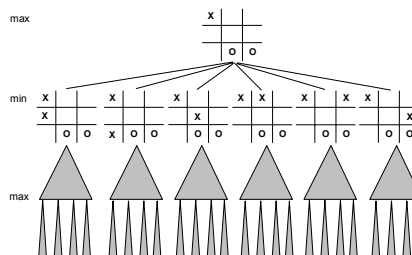
13



## Παράδειγμα Εφαρμογής Min-Max στη Τρίλιζα (i)

14

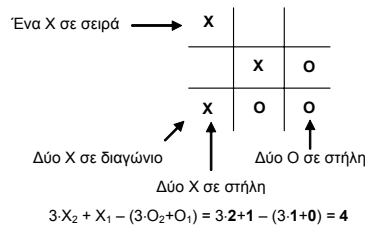
- Το παίγνιο τρίλιζα είναι ένα πρόβλημα με μικρό χώρο αναζήτησης, της τάξης των 9! καταστάσεων.
- Όλες οι καταστάσεις μπορεί κάλλιστα να κωδικοποιηθούν σε κάποιο πρόγραμμα το οποίο σαν αποτέλεσμα δε θα χάνει ποτέ.
- Αποτελεί καλό παράδειγμα εφαρμογής του αλγορίθμου Minimax.
- Έστω ότι ο παίκτης Max είναι ένα πρόγραμμα που σημειώνει X και αντιμετωπίζει έναν άνθρωπο ή ένα άλλο πρόγραμμα, τον Min.
- Έστω ότι ο Max αποφασίζει να δει μέχρι βάθος τριών κινήσεων.
- Στο βάθος αυτό εφαρμόζει τη συνάρτηση αξιολόγησης για όλες τις καταστάσεις που έχουν προκύψει.



## Παράδειγμα Εφαρμογής Min-Max στη Τρίλιζα (ii)

15

- Μία συνάρτηση αξιολόγησης στην τρίλιζα θα μπορούσε να είναι η:
 
$$3 \cdot X_2 + X_1 - (3 \cdot O_2 + O_1)$$
  - $X_2$  ο αριθμός γραμμών, στηλών ή διαγωνίων με δύο X και χωρίς κανένα O.
  - $X_1$  ο αριθμός γραμμών, στηλών ή διαγωνίων με ένα X και χωρίς κανένα O.
  - $O_2$  ο αριθμός γραμμών, στηλών ή διαγωνίων με δύο O και χωρίς κανένα X.
  - $O_1$  ο αριθμός γραμμών, στηλών ή διαγωνίων με ένα O και χωρίς κανένα X.
- Προφανώς, θετικές τιμές επιστρέφουν οι καταστάσεις που ωφελούν αυτόν που σημειώνει X, ενώ αρνητικές τιμές αυτές που ωφελούν αυτόν που σημειώνει O.
- Για παράδειγμα, στο επόμενο σχήμα απεικονίζεται μία κατάσταση της οποίας η συνάρτηση αξιολόγησης επιστρέφει 4, δηλαδή αυτός που σημειώνει το X έχει εμφανές πλεονέκτημα.



## Παράδειγμα Εφαρμογής Min-Max στο Σκάκι (i)

16

- Το κύριο μέλημα των προγραμμάτων σκάκι είναι να αναζητήσουν το δένδρο του παιχνιδιού σε όσο το δυνατόν μεγαλύτερο βάθος. Αν υπήρχε η δυνατότητα να φτάσουν μέχρι τις τελικές καταστάσεις του παιχνιδιού, τότε τα προγράμματα θα ήταν ανίκητα.
- Μία συνάρτηση αξιολόγησης στο σκάκι
  - **Υπεροχή κομματιών:** Κάθε κομμάτι έχει κάποια αξία, π.χ. Βασιλιάς=10, Άλογο=5, Πιόνι=1 κλπ. Η αξία όλων των κομματιών κάθε χρώματος προστίθεται.
  - **Υπεροχή θέσης:** Κάθε κομμάτι που βρίσκεται στα 4 κεντρικά τετράγωνα παίρνει επιπλέον 2 πόντους.
  - **Απειλές:** Για κάθε απειλή που προβάλλει ένας παίκτης παίρνει 3 επιπλέον πόντους, εκτός αν απειλεί το βασιλιά του άλλου παίκτη, οπότε παίρνει 20 πόντους.



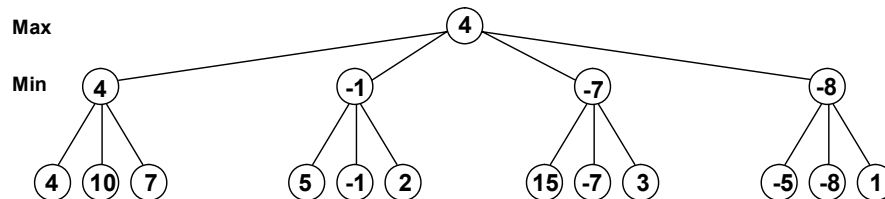


## Σύγκριση του AB με τον Min-Max (i)



19

### Παράδειγμα Min-Max

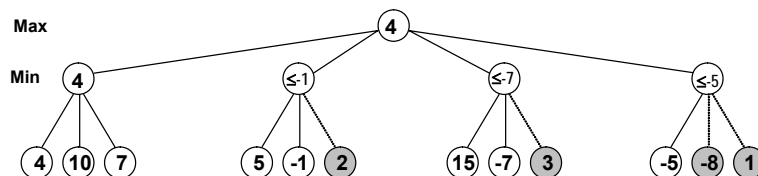


## Σύγκριση του AB με τον Min-Max (ii)



20

### Παράδειγμα Alpha-Beta



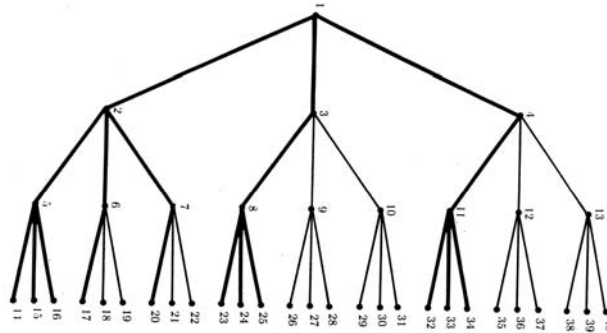
#### Εξηγήσεις

- Αξιολογούνται οι πρώτοι αριστεροί κόμβοι (4, 10, 7) και δίνουν την τιμή 4 στον κόμβο πατέρα.
- Η αξιολόγηση προχωρά στο δεύτερο κλαδί και βρίσκει 5 και -1 οπότε και σταματά γιατί η τιμή του είναι μικρότερη από τη μεγαλύτερη τιμή του ανωτέρου επιπέδου (δηλ. το 4).
- Όμοια γίνεται και στους επόμενους κόμβους.

## Παράδειγμα Alpha-Beta



21



## Σύγκριση του AB με τον Min-Max (ii)



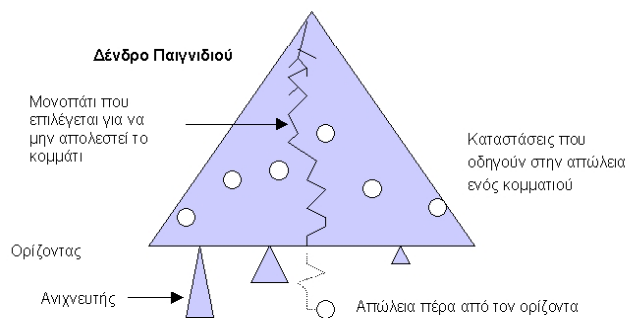
22

- Κατά προσέγγιση ο AB εξετάζει τερματικούς κόμβους, όπου  $N$  είναι οι τερματικοί κόμβοι που εξετάζει ο αλγόριθμος Minimax.
- Η απόδοσή του βελτιώνεται με διάφορες μεθόδους, όπως:
  - Ευριστικό κλάδεμα του δένδρου παιχνιδιού.
  - Δυναμική αντί στατικής συνάρτησης αξιολόγησης.
  - Αποθήκευση τιμών των τερματικών καταστάσεων (transposition tables).
  - Προκαθορισμένες κινήσεις (χωρίς αναζήτηση) σε αρχικές και τελικές φάσεις του παιχνιδιού (*Openings, End Game moves*).

# Το Πρόβλημα του Ορίζοντα



23



## Μειονεκτήματα:

Το φαινόμενο του ορίζοντα (horizon effect). Αντιμετωπίζεται με ανιχνευτές (scouts).