بسمه تعالی

هوش مصنوعی

جستجوی تخاصمی و بازیها

نیمسال اوّل ۱۴۰۴-۱۴۰۳

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکدهٔ مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

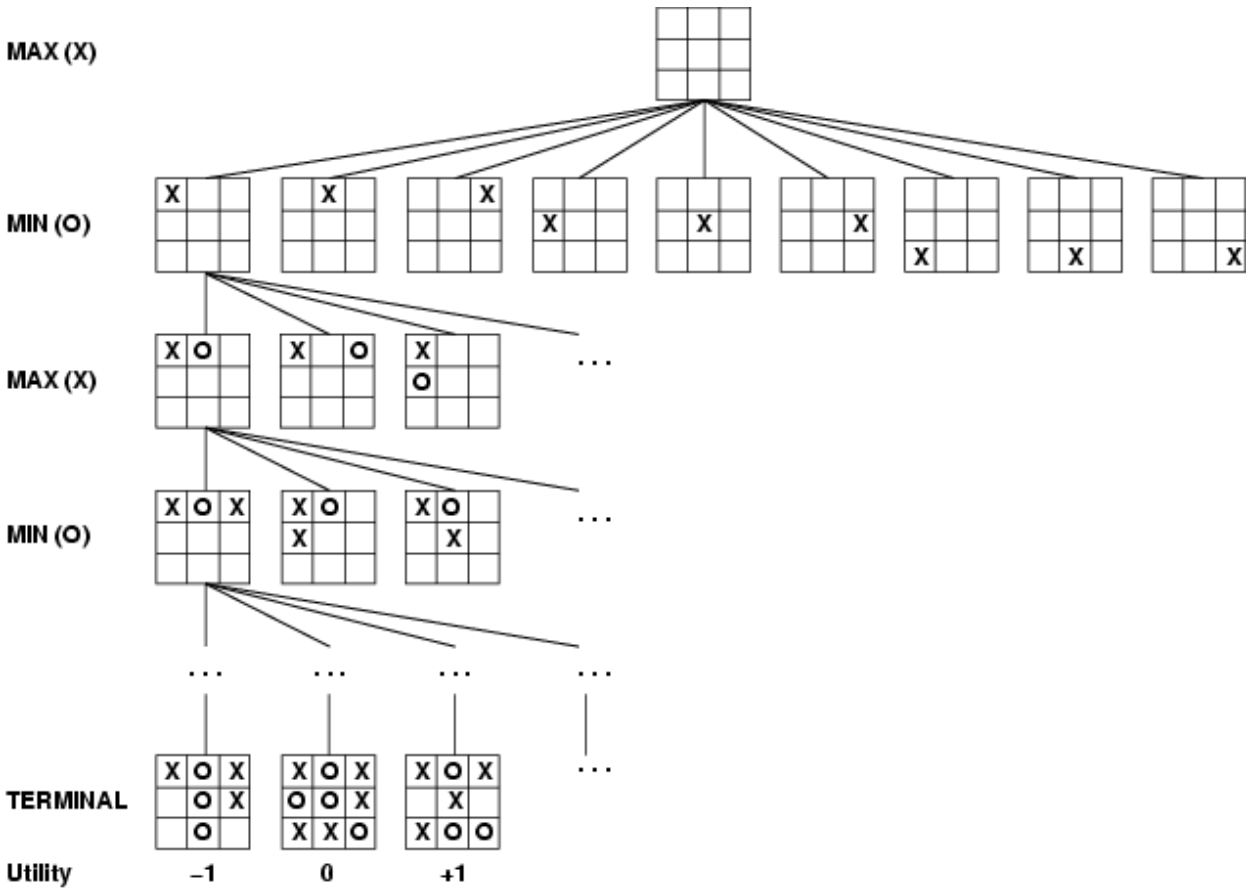
مقدمه

- در محیطهای چندعاملی، عامل ممکن است همکار بوده یا با هم در رقابت باشند.
- هنگامی که عاملها با هم در رقابت هستند، یعنی اهداف متضادی دارند، جستجوی تخاصمی که به آن بازی نیز گفته می شود استفاده می شود.
- از ابتدای پیدایش کامپیوتر

بازی دو نفره

- بازی دو نفره هر یک به نوبت
- تعریف به عنوان یک مسئلهٔ جستجو
- حالت اولیه: وضعیت تخته S_0
- نوبت کدام بازیکن To_Move(s)
- اعمال مجاز در حالت s: Actions(s)
- مدل انتقال Result(s،a): نتیجهٔ انجام عمل a در حالت s
- تست هدف: تعیین خاتمهٔ بازی Is_Terminal(s)
  - حالاتی که بازی در آنها خاتمه می یابد حالات پایانی (ترمینال)
- ارزیابی: امتیاز به وضعیت نهائی بازی Utility(s،p)
  - مقدار عددی نهائی بازی با حالت پایانی s برای بازیکن p

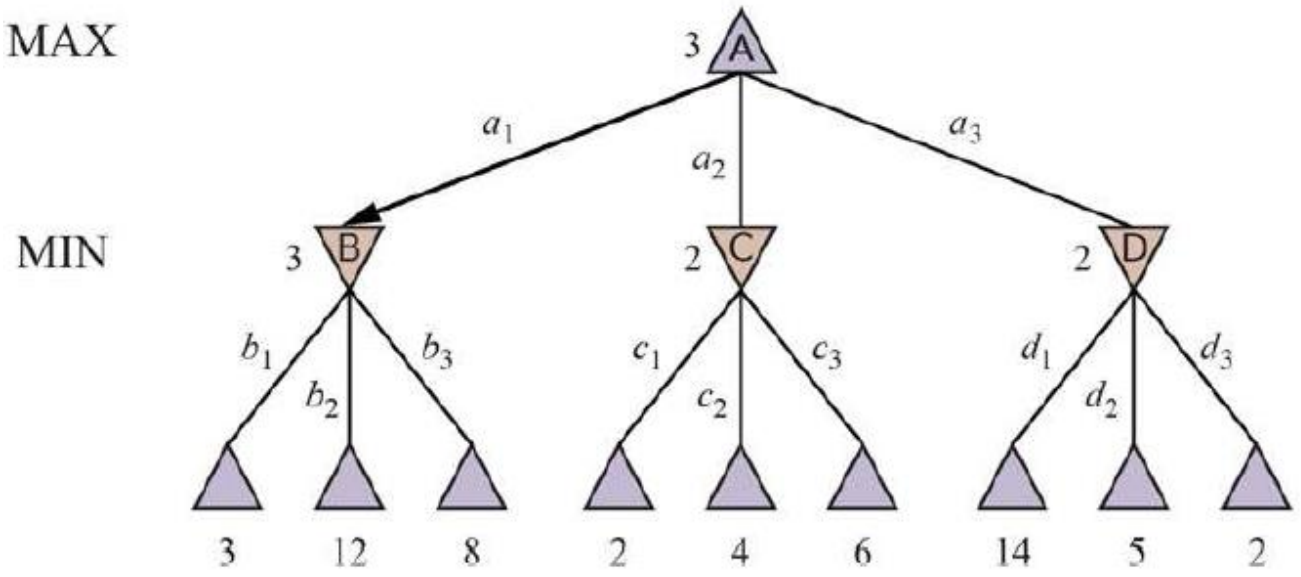
- برای شطرنج نتیجهٔ بازی برد، باخت، یا مساوی (۱، ۱-، ۰)
- بازی جمع-صفر zero-sum game به بازی گفته می شود که جمع امتیاز بازیکنان برابر صفر شود.
- برای شطرنج ۰+۰=1+1-=1-+1
- با اغماض به بازیهائی که جمع امتیاز بازیکنان همواره برابر شود.
- اگر امتیازها بصورت (۱و ۰و ۱/۲) باشد جمع همواره یک است.
- حالت اولیه و حرکات مجاز یک درخت بازی ایجاد می کنند.


MAX (X)
MIN (O)
MAX (X)
MIN (O)
TERMINAL
Utility
–1
0
+1

- MAX باید تمام حرکات MIN را در نظر بگیرد.
- به نوعی یک مسئلهٔ اقتضائی است.
- برای بازیهائی که نتایج دوتائی (برنده-بازنده) دارند، می توان جستجوی AND-OR را برای تولیدطرح شرطی استفاده کرد.
- در واقع همانند وضعیت غیرقطعی می باشد.
- برای بازیهائی با نتایج بیشتر جستجوی minimax را می توان استفاده نمود.

Minimax

- انتخاب حرکتی که بیشترین مقدار کمینه–بیشینه را دارد.



$$\textsc{Minimax}(s) = \begin{cases} \textsc{Utility}(s, \textsc{max}) & \text{if } \textsc{Is-Terminal}(s) \\ \max_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{To-Move}(s) = \textsc{max} \\ \min_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s, a)) & \text{if } \textsc{To-Move}(s) = \textsc{min} \end{cases}$$

```
function MINIMAX-SEARCH(game, state) returns an action
    player ← game.TO-MOVE(state)
    value, move ← MAX-VALUE(game, state)
    return move
```

```
function MINIMAX-SEARCH(game, state) returns an action
    player ← game.TO-MOVE(state)
    value, move ← MAX-VALUE(game, state)
    return move

function MAX-VALUE(game, state) returns a (utility, move) pair
    if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
    v ← −∞
    for each a in game.ACTIONS(state) do
        v2, a2 ← MIN-VALUE(game, game.RESULT(state, a))
        if v2 > v then
            v, move ← v2, a
    return v, move
```

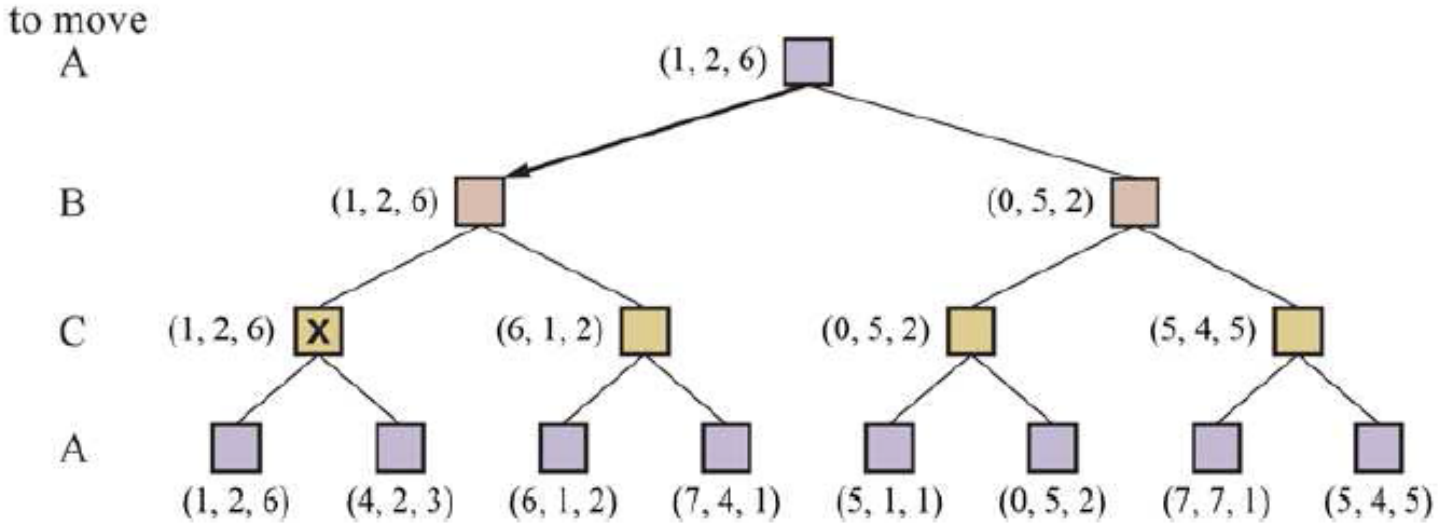
```
function MINIMAX-SEARCH(game, state) returns an action
  player ← game.TO-MOVE(state)
  value, move ← MAX-VALUE(game, state)
  return move

function MAX-VALUE(game, state) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← −∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MIN-VALUE(game, game.RESULT(state, a))
    if v2 > v then
      v, move ← v2, a
  return v, move

function MIN-VALUE(game, state) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← +∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MAX-VALUE(game, game.RESULT(state, a))
    if v2 < v then
      v, move ← v2, a
  return v, move
```

- کامل: بله (اگر درخت محدود باشد)
- بهینه: بله (درمقابل حریف بهینه)
- اگر حریف غیربهینه باشد، MAX حداقل همانند حریف بهینه عمل می کند.
- با فرض m حداکثر عمق درخت و
- b حرکات مجاز در هر نقطه
- پیچیدگی زمانی: $O(b^m)$
- پیچیدگی فضا:
  - اگر همهٔ حرکات بسط داده شوند $O(bm)$
  - اگر فقط یکی از حرکات بسط داشته شود $O(m)$

- چون پیچیدگی زمانی نمائی است برای بازیهای پیچیده غیرعملی است.
- برای شطرنج b حدود ۳۵ و m حدود ۸۰
- مقدار $35^{80} \approx 10^{123}$ غیر عملی

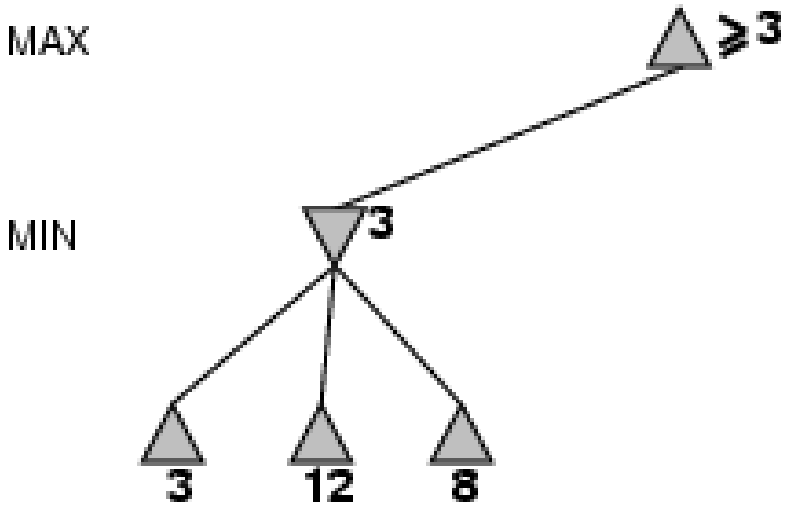
بازی با چند بازیکن

Figure 5.4

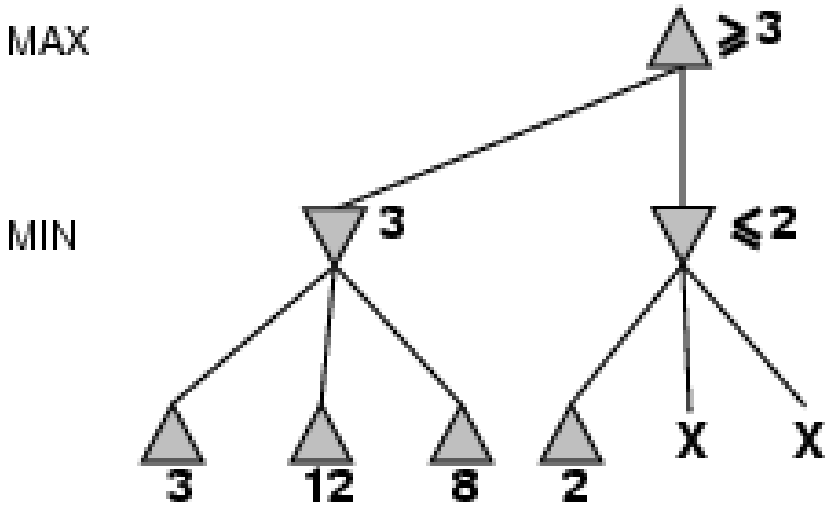


The first three ply of a game tree with three players (A, B, C). Each node is labeled with values from the viewpoint of each player. The best move is marked at the root.

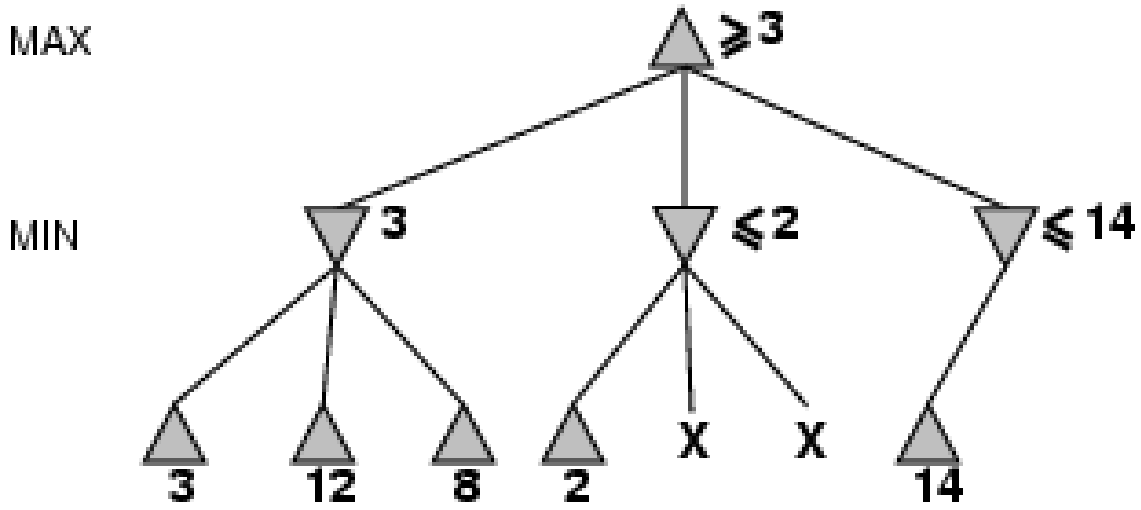
هرس α–β



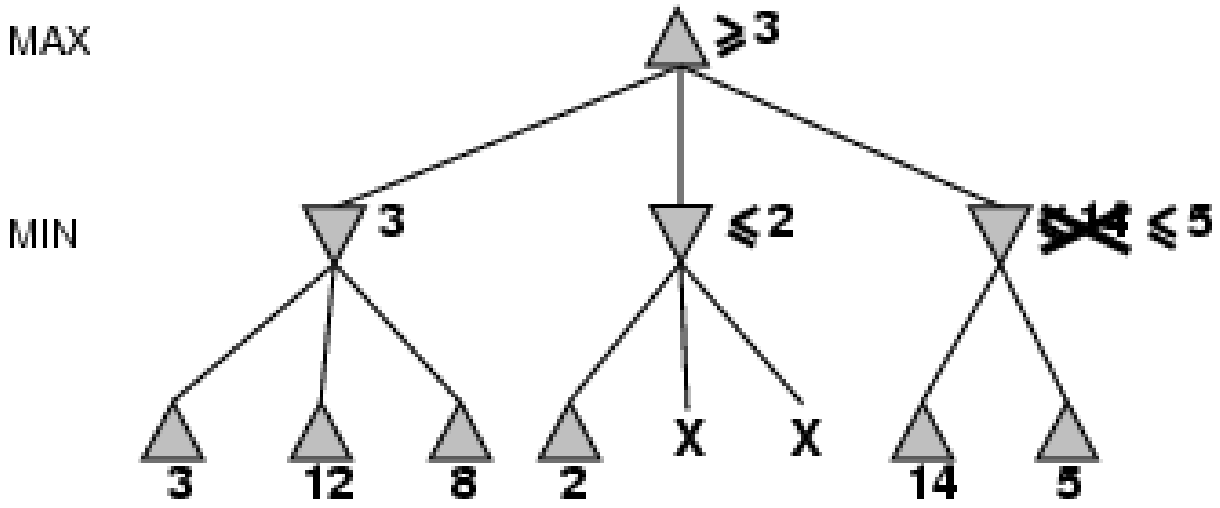
هرس α–β



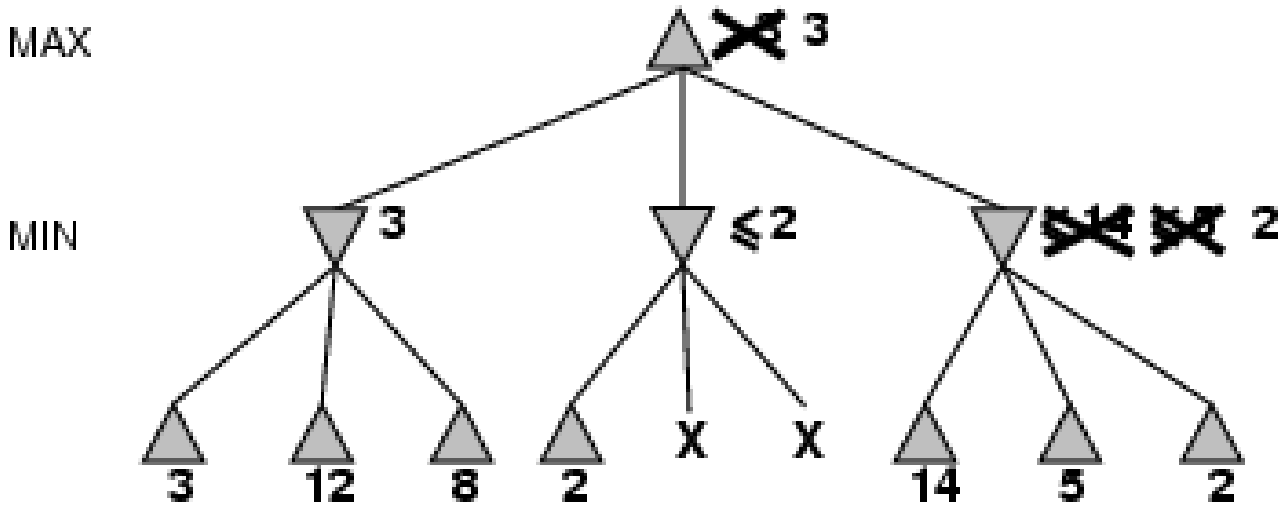
هرس α–β

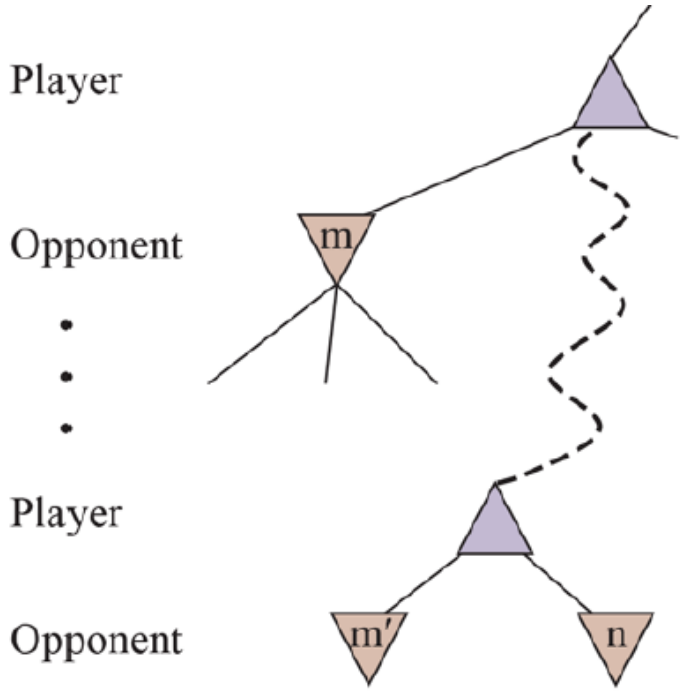


α–β هرس



هرس α–β



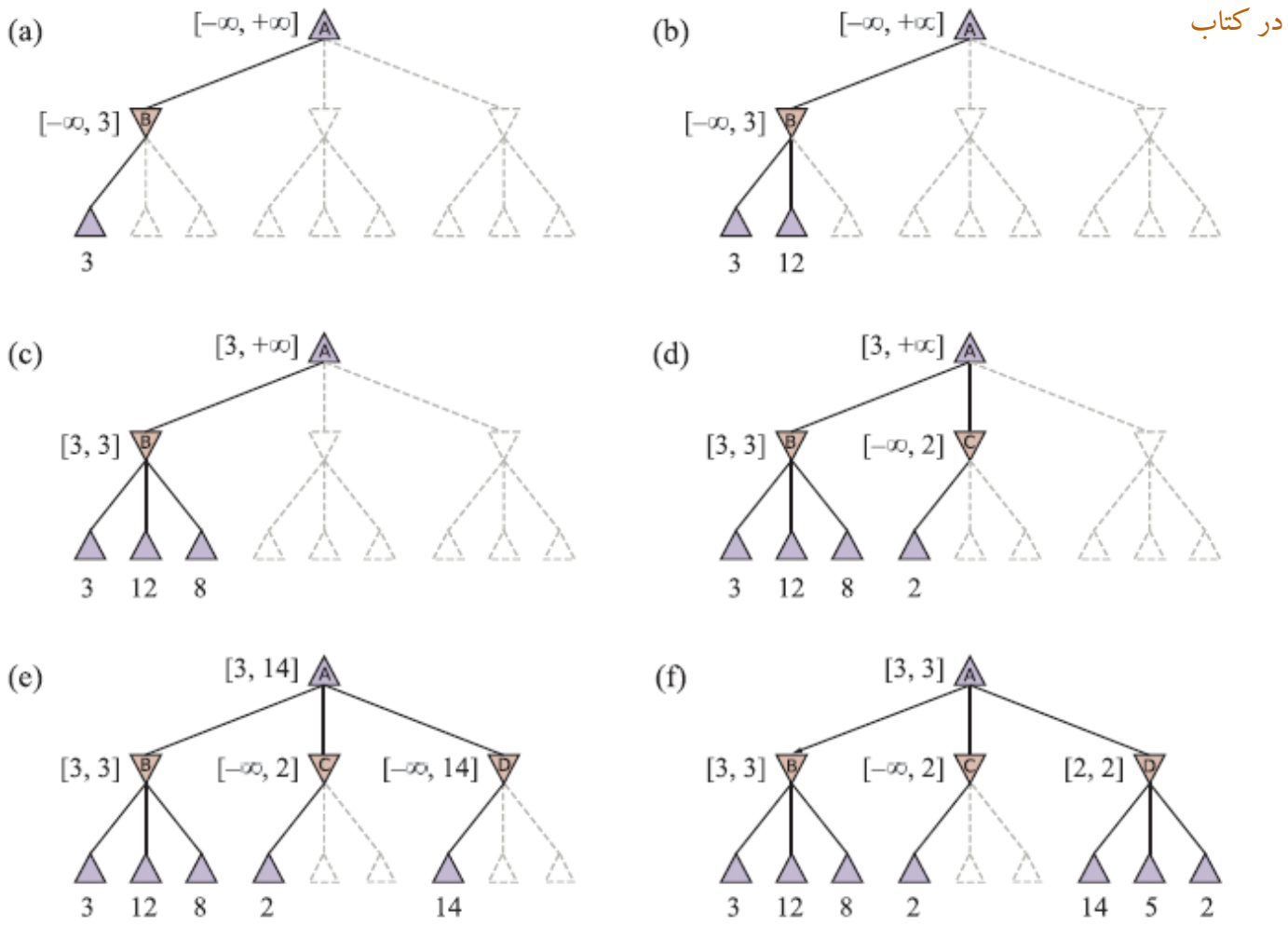


- عمل هرس در هر عمقی ممکن است رخ دهد.
- اگر m یا m’ بهتر ا ز n برای player باشند، در بازی هیچگاه به n نخواهیم رسید.
- هنگامی که اطلاعات کافی در مورد n بدست آمد می توان آن را هرس کرد.

هرس α-β

- α مقدار بهترین انتخاب (بالاترین مقدار) که در مسیر MAX تاکنون یافته شده است.
- تصور کنید α= حداقل
- β مقدار بهترین انتخاب (کمترین مقدار) که در مسیر MIN تاکنون یافته شده است.
- تصور کنید β= حداکثر

در کتاب


(a)
[−∞, +∞]
A
[−∞, 3]
B
3
(b)
[−∞, +∞]
A
[−∞, 3]
B
3 12
(c)
[3, +∞]
A
[3, 3]
B
3 12 8
(d)
[3, +∞]
A
[3, 3]
B
[−∞, 2]
C
3 12 8 2
(e)
[3, 14]
A
[3, 3]
B
[−∞, 2]
C
[−∞, 14]
D
3 12 8 2 14
(f)
[3, 3]
A
[3, 3]
B
[−∞, 2]
C
[2, 2]
D
3 12 8 2 14 5 2

```
function ALPHA-BETA-SEARCH(game, state) returns an action
    player ← game.TO-MOVE(state)
    value, move ← MAX-VALUE(game, state, −∞, +∞)
    return move
```

```
function ALPHA-BETA-SEARCH(game, state) returns an action
  player ← game.TO-MOVE(state)
  value, move ← MAX-VALUE(game, state, −∞, +∞)
  return move

function MAX-VALUE(game, state, α, β) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← −∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MIN-VALUE(game, game.RESULT(state, a), α, β)
    if v2 > v then
      v, move ← v2, a
      α ← MAX(α, v)
    if v ≥ β then return v, move
  return v, move
```

اصلاح حداقل مقدار

اگر کمترین مقداری که Max برمی گرداند از امتیاز کنونی Min بالاسر بیشتر است بیشتر جستجو نکن

```
function ALPHA-BETA-SEARCH(game, state) returns an action
  player ← game.TO-MOVE(state)
  value, move ← MAX-VALUE(game, state, −∞, +∞)
  return move

function MAX-VALUE(game, state, α, β) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← −∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MIN-VALUE(game, game.RESULT(state, a), α, β)
    if v2 > v then
      v, move ← v2, a
      α ← MAX(α, v)
    if v ≥ β then return v, move
  return v, move

function MIN-VALUE(game, state, α, β) returns a (utility, move) pair
  if game.IS-TERMINAL(state) then return game.UTILITY(state, player), null
  v ← +∞
  for each a in game.ACTIONS(state) do
    v2, a2 ← MAX-VALUE(game, game.RESULT(state, a), α, β)
    if v2 < v then
      v, move ← v2, a
      β ← MIN(β, v)
    if v ≤ α then return v, move
  return v, move
```

اصلاح حداقل مقدار

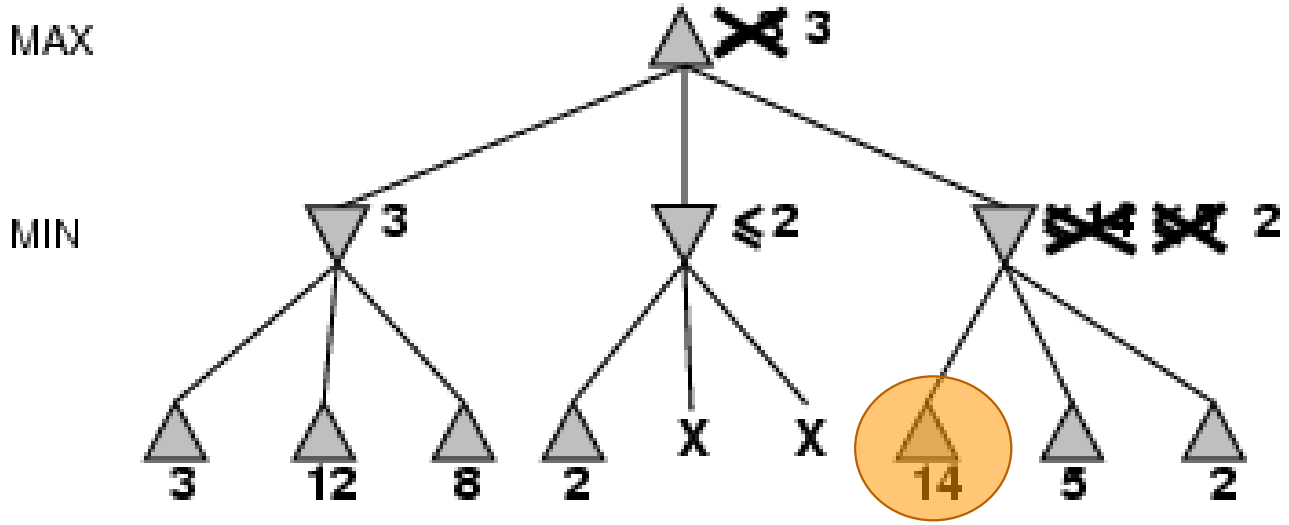
اگر کمترین مقداری که Max برمی گرداند از امتیاز کنونی Min بالاسر بیشتر است بیشتر جستجو نکن

اصلاح حداکثر مقدار

اگر حداکثر مقداری که Min برمی گرداند از امتیاز کنونی Max بالاسر کمتر است بیشتر جستجو نکن

25

- کارآئی قطع آلفا-بتا تا حد زیادی وابسته به ترتیبی است که تالیها را در نظر می گیریم.




پایان

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائهٔ شفاهی می باشد و **به هیچ وجه یک جزوهٔ درسی نیست** و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.