

دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

گزارش ۴

پیاده سازی یک بازی دو نفره با قابلیت‌های جستجوی تخصصی
(بازی Tic-Tac-Toe با الگوریتم α - β pruning)

نام و نام خانوادگی : محمدحسین ملک‌پور

رشته تحصیلی : علوم کامپیوتر

شماره دانشجویی : ۹۶۱۳۴۲۵

استاد : دکتر مهدی قطعی

پست الکترونیکی : mohammadhossein.malekpour@gmail.com

فهرست

۳ alpha-beta pruning + minimax algorithm	الگوریتم
۵ alpha-beta pruning	پایاده سازی بازی Tic-Tac-Toe با الگوریتم
۷	منابع

الگوریتم α - β pruning + minimax algorithm

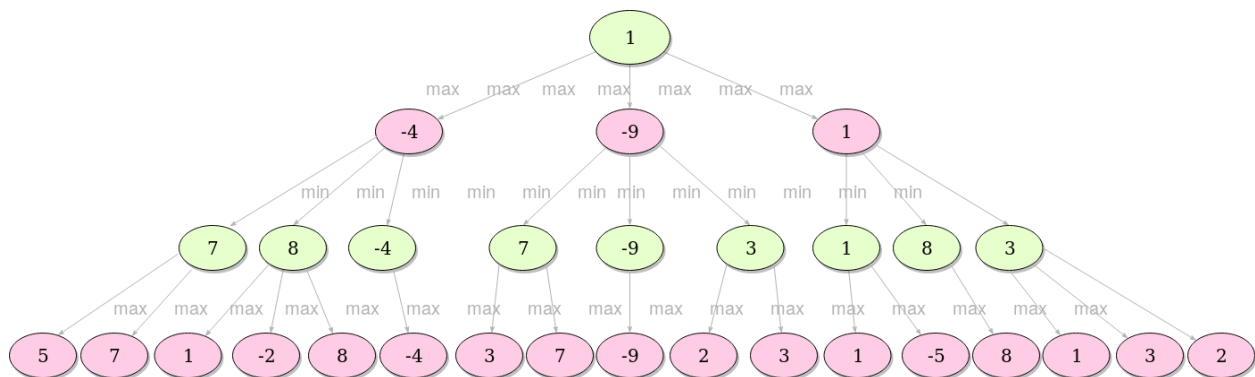
الگوریتم مینیماکس (minimax algorithm) روشی برای یافتن حرکت بهینه در بازی دونفره است. هرس آلفا-بتا (α - β pruning) روشی است برای یافتن جواب بهینه‌ی الگوریتم مینیماکس که از جست‌وجو در برخی فضاهایی که انتخاب نمی‌شوند، جلوگیری می‌کند. به این ترتیب جست‌وجو در درخت کمینه-بیشینه تا عمق مشخصی در زمان کمتری انجام می‌شود.

برای بررسی ایده کلی هرس آلفا بتا این دو مسئله مشابه را در نظر بگیرید:

– تعدادی زیر مجموعه ناتهی و متناهی از مجموعه اعداد حقیقی در اختیار داریم. ارزش (یا امتیاز) هر یک از این مجموعه‌ها را برابر با کوچک‌ترین عضو آن تعریف می‌کنیم. هدفمان یافتن مجموعه‌ای با بیش‌ترین ارزش است. فرض کنید که ارزش یکی از مجموعه‌ها برابر با m است. در این صورت مجموعه‌ای که دست کم یک عضو کوچک‌تر از m داشته باشد، پاسخ مسئله نخواهد بود. پس نیازی به بررسی اعضای این مجموعه (و یافتن ارزش آن) نیست. (چرا که ارزش آن کوچک‌تر از m است)

– همان پرسش بالا را این گونه تغییر می‌دهیم: ارزش هر مجموعه برابر با بزرگ‌ترین عضو آن تعریف می‌شود و هدف یافتن کم ارزش‌ترین مجموعه است. در این حالت نیز اگر مجموعه‌ای با ارزش m وجود داشته باشد مجموعه‌هایی که حداقل یک عضو بزرگ‌تر از m دارند، نمی‌توانند پاسخ مسئله باشند.

مثال:



در درخت کمینه- بیشینه یا درخت بازی در الگوریتم مینیماکس به صورت جست و جوی عمقی ساخته می‌شود. در روش آلفا بتا، به هر راس درخت کمینه- بیشینه دو مقدار α و β به صورت زیر اختصاص می‌دهیم:

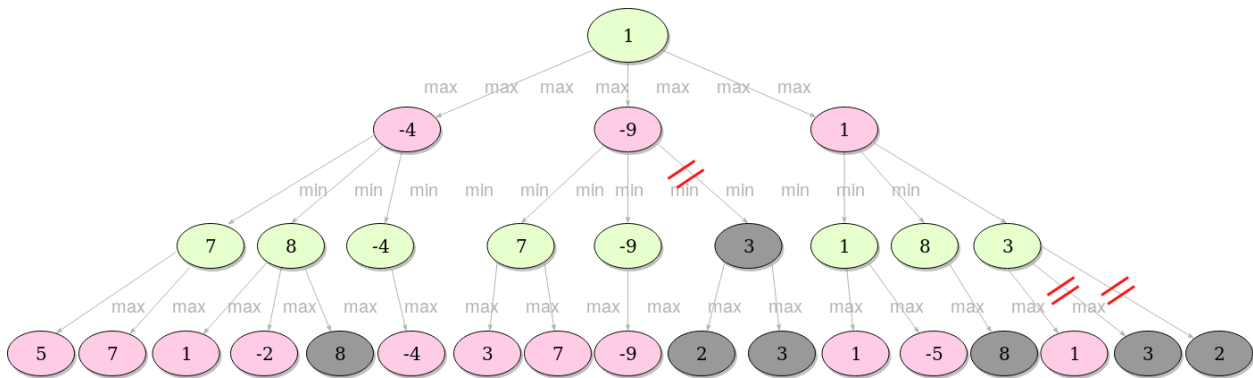
- مقدار α : بیشینه حد پایین امتیازی که بین راه‌حل‌ها به آن رسیده‌ایم.
- مقدار β : کمینه حد بالای امتیازی که بین راه‌حل‌ها به آن رسیده‌ایم.

بنابراین به هنگام پیمایش، جست‌وجو در زیردرخت راسی را ادامه می‌دهیم که شرط زیر را دارا باشد:

$\alpha \leq N \leq \beta$ که عدد N تخمین فعلی از مقدار آن راس است.
نحوه‌ی بروزرسانی در طی پیمایش:

- راس‌های کمینه در درخت کمینه- بیشینه مقدار α را تغییر می‌دهند.
- راس‌های بیشینه در درخت کمینه- بیشینه مقدار β را تغییر می‌دهند.

مثال:



پیاده سازی بازی Tic-Tac-Toe با الگوریتم alpha-beta pruning

در این [لینک](#) (گیتهاب) به کد بازی دسترسی خواهید داشت.

فانکشن های کلاس TicTacToe بصورت زیر عمل میکند:

`initialize_game()` : ساخت برد بازی

`draw_board()` : پرینت کردن وضعیت فعلی برد بازی در ترمینال

`is_valid()` : بررسی امکان پذیر بودن حرکت انجام شده توسط کاربر

`is_end()` : بررسی وضعیت بازی که اگر به پایان رسیده برنده را پیدا کند

`max_alpha_beta()`: بازیکن O هوش مصنوعی که در مقابل آن بازی می کنیم به دنبال دو چیز است، به حداکثر رساندن امتیاز خود و به حداقل رساندن امتیاز ما. و اینگونه تصمیم بهینه را میگیرد.

`min_alpha_beta()`: بازیکن X ما (انسان) هستیم و این تابع به ما کمک میکند سود هوش مصنوعی را کمینه کنیم.

`alpha_beta_play()`: حلقه تکرار اصلی بازی که از توابع بالا در آن استفاده میکنیم.

و در نهایت یک شی از کلاس فوق میسازیم و بازی را با تابع `play_alpha_beta` اجرا میکنیم.

در صفحه ی بعد کد را اجرا کردیم و با ریکامندهای هوش مصنوعی پیش رفتیم و بازی مساوی شد.

```
PS C:\Users\ASUS\Desktop\ex5-9613425> & 'C:\Users\ASUS\AppData
10160' '--' 'c:\Users\ASUS\Desktop\ex5-9613425\code.py'
```

```
.| .| .|
.| .| .|
.| .| .|
```

Recommended move: X = 0, Y = 0

Insert the X coordinate: 0

Insert the Y coordinate: 0

```
X| .| .|
.| .| .|
.| .| .|
```

```
X| .| .|
.| O| .|
.| .| .|
```

Recommended move: X = 0, Y = 1

Insert the X coordinate: 0

Insert the Y coordinate: 1

```
X| X| .|
.| O| .|
.| .| .|
```

```
X| X| O|
.| O| .|
.| .| .|
```

Recommended move: X = 2, Y = 0

Insert the X coordinate: 2

Insert the Y coordinate: 0

```
X| X| O|
.| O| .|
X| .| .|
```

```
X| X| O|
O| O| .|
X| .| .|
```

Recommended move: X = 1, Y = 2

Insert the X coordinate: 1

Insert the Y coordinate: 2

```
X| X| O|
O| O| X|
X| .| .|
```

```
X| X| O|
O| O| X|
X| O| .|
```

Recommended move: X = 2, Y = 2

Insert the X coordinate: 2

Insert the Y coordinate: 2

```
X| X| O|
O| O| X|
X| O| X|
```

It's a tie!

```
PS C:\Users\ASUS\Desktop\ex5-9613425> █
```

منابع:

https://en.wikipedia.org/wiki/Alpha%E2%80%A3beta_pruning

https://en.wikipedia.org/wiki/Minimax#Minimax_algorithm_with_alternate_moves

<https://stackabuse.com/minimax-and-alpha-beta-pruning-in-python/>