به نام خدا

پروژه درس هوش مصنوعی
پروژه بازی - الگوریتمminimax

محمدرضا شش پری
810198417

بهار 1401

هدف انجام پروژه:

آشنایی با الگوریتم minimax و استفاده از آن برای رقابت در بازی connect-4

توضیح کلی مساله:

در این بازی دو agent داریم که ما وظیفه کنترل یکی از آنها را داشته و مامور دیگر توسط کامپیوتر هدایت می شود. باید در هر مرحله یک ستون از ماتریکس صفحه بازی را انتخاب ، و با انداختن مهره ای در آخرین خانه خالی آن ستون، به هدف اصلی بازی که رسیدن به 4 خانه متوالی همرنگ در یکی از 3 جهت (افقی، عمودی و یا مورب) است ، برسیم.

همچنین می دانیم که اگر ستونی پر باشد، نمیتوانیم در آن مهره ای بیندازیم.

برای انتخاب ستون مورد نظر، باید از الگوریتم minimax استفاده کنیم و از یک huristic نیز برای بازی استفاده کنیم.

مدل کردن مساله:

کلیت مساله و قوانین آن از قبل پیاده سازی شده و کد آن در اختیار ما قرار داده شده است. ما صرفا باید تابعی که مربوط به انتخاب خانه توسط agent خودمان است را پیاده سازی کنیم.

پیاده سازی:

هیوریستیک: در ابتدا برای هر دو بازیکن تعداد خانه هایی که connect2 و connect3 بوده (در سه راستای مورب،افقی و عمودی) و خانه مجاور یکی از دو سر آن ها خالی است را بدست آورده و سپس به ترتیب به ازای هر کدام از آنها امتیاز 1 و 4 دادیم. (در صورتی که خانه های متصل برای ما بود مثبت 1و4 و در صورتی که خانه ها برای حریف بود، منفی 1و4 دادیم.) همچنین اگر connect4 وجود داشت امتیاز 1000 دادیم.

Minimax :در این تابع که براساس الگوریتم هم نام خودش است، ما دو حالت کلی داریم.

در صورتی که به عمق موردنظر (ورودی کاربر) رسیده باشیم، از تابع هیوریستیک برای تخمین جدول استفاده می کنیم.

در غیراینصورت نیز برای تمامی فرزندان جدول (state) جاری (منظور از فرزندان، تمام حالت هایی است که از افزودن 1 مهره به صفحه جاری بازی بدست می آید)تابع را بصورت بازگشتی با یک عمق بیشتر و جدول جدید(جدولی که یک مهره جدید به آن اضافه شده) و نوبتی مخالف با نوبت بازیکن جاری صدا می زنیم.سپس در صورتی که نوبت ما بود،خانه حاصل از max خروجی تابع و درصورتی که نوبت حریف بود، min حاصل از هیوریستیک خروجی تابع را بعنوان خانه جواب خروجی می دهیم.

برای حالت alpha-beta نیز بصورت الگوریتم minimax عمل میکنیم، فقط درقسمتی که مقدار min/max را بدست می آوریم، شرط الگوریتم alpha-beta را بررسی می کنیم و درصورت ارضای شرط، از بررسی بقیه فرزندان آن عمق خودداری می کنیم.

جدول خروجی نتایج بازی :

Alpha-Beta (200 time)								
row	Board size	Depth	Average time(msec)	Total time (sec)	Avg. num of visited nodes	Number of wins	Number of failures	Win rate (%)
1	6*7	1	14	2.85	72	175	25	87.5
2	6*7	3	71	14.2	350	187	13	93.5
3	6*7	5	2710	543	16500	188	12	94
4	6*7	7	130000	26000	650000	190	10	95
5	7*8	1	20	4.1	79	172	28	86
6	7*8	3	129	25.9	470	197	3	98.5
7	7*8	5	7275	1820	26000	196	4	98
8	7*8	7	385000	77000	1480000	197	3	98.5
9	7*10	1	36	7.2	90	179	21	89.5
10	7*10	3	27	54	770	199	1	99.5
11	7*10	5	23000	4600	75000	199	1	99.5
12	7*10	7	1378000	275600	4444000	199	1	99.5

Minimax (200 time)								
row	Board size	Depth	Average time(msec)	Total time (sec)	Avg. num of visited nodes	Number of wins	Number of failures	Win rate (%)
1	6*7	1	60	12	300	186	14	93
2	6*7	3	393	78.6	2000	196	4	98
3	6*7	5	21000	4200	120000	196	4	98
4	7*8	1	103	20.6	400	178	22	89
5	7*8	3	850	170	3000	198	2	99
6	7*8	5	63500	12700	260000	197	3	98.5
7	7*10	1	200	40	650	182	18	91
8	7*10	3	1990	398	5700	196	4	98
9	7*10	5	218000	43600	700000	197	3	98.5

همانطور که در جدول می بینیم؛ بصورت کلی با افزایش عمق، زمان اجرای برنامه و تعداد حالت های دیده شده و نرخ پیروزی افزایش پیدا می کند.

همچنین الگوریتم alpha-beta نسبت به الگوریتم minimax ساده، حالت های کمتری را بازدید می کند که نتیجه آن افزایش سرعت اجرای آن خواهد بود.

پاسخ سوالات:

1*)
بطور کلی هیوریستیکی خوب است که برای استیتی که به جواب مساله نزدیک تر (در مقایسه با استیت های دیگر) است، عددی که نشان دهنده بهتر بودن جواب است را خروجی می دهد، به عبارت دیگر تخمین آن از شرایط موجود، معیار دقیقی ازاستیت می باشد. از ویژگی های دیگر آن نیز می توان به سریع و آسان بودن امکان محاسبه آن اشاره کرد.

هیوریستیک استفاده شده:

در توضیحات بالا گفته شد.

علت خوب بودن این تابع تخمین، استفاده از معیار های درست (connect2,connect3)در برنده شدن در بازی برای تخمین زدن است.

روش های دیگر:

تعداد خانه های پشت سر هم (در نظر نگرفتن خانه خالی در مجاورت دو سر connect-i) : این تخمین از تابع استفاده شده کم دقت تر است چون مثلا به خانه هایی که connect3 هستند ولی دو خانه همسایه آن توسط مهره دیگری پر است، امتیاز مثبت می دهد در حالی که به وضوح این خانه ها برای connect4 غیرقابل استفاده است.

همچنین می توان هیوریستیک های دیگری مثل در نظر گرفتن تعداد خانه های connect2,connect3 بدون ضریب و یا تعداد همین خانه ها با ضرایب 1و4 ولی فقط در راستای افقی (لحاظ نکردن راستای عمودی و مورب) که این توابع نیز به وضوح تخمین ضعیف تری نسبت به مورد گفته شده ارائه می کنند.

2*)بطور کلی اگرعمق بررسی ما بیشتر باشد، شانس پیروزی بالاتر و زمان و تعداد گره های دیده شده ، بیشتر خواهد بود. همچنین به یک مورد باید توجه کرد و آن مورد این است که در این الگوریتم ما فرض می کنیم که حریف ما در هر مرحله بهترین خانه را انتخاب می کند و بر این اساس تصمیم میگیریم ، اما از آنجایی که حریف در این الگوریتم بهینه عمل نمی کند (رندم عمل می کند) الزاما بیشتر بودن عمق ، شانس پیروزی بیشتری را به ارمغان نمی آورد وهمانطور که دیدیم، ممکن است یک عمق کمتر ، شانس پیروزی بیشتری داشته باشد.

3*)درالگوریتم پیاده سازی شده، فرزندان به ترتیب از چپ ترین ستون به راست ترین ستون اضافه می شوند . علت انتخاب این ترتیب ساده تر بودن پیاده سازی آن و پرهیز از درگیر شدن با حالت های مختلف برای انتخاب یک فرزند است .مشخصا در روش هرس،ترتیب افزودن فرزندان تاثیر گذار است(بخاطر شرط قطع کردن درخت در صورت وقوع شرط حرس آلفا بتا)

همچنین از روش های بهتر برای ترتیب افزودن فرزندان می توان به بررسی ستون هایی که مهره های بیشتری در آنها یا ستون های مجاور آنها وجود دارد، اشاره کرد.