



دانشگاه تهران
دانشکده ریاضی، آمار، علوم کامپیوتر

حسین نظری ۶۱۰۳۹۸۱۷۹
گزارش تمرین ۲ درس هوش مصنوعی

-تعریف state: در این مساله هر استیت نشان دهنده هر حالت قرارگیری ۹ عدد ۰ تا ۸ کنار یکدیگر در یک ماتریس 3×3 می باشد. برای سادگی به جای ماتریس 2 بعدی از ابعاد 3×3 می توان آن را با یک ماتریس 1×9 نیز نمایش داد. برای مثال داریم: $[1,2,4,6,0,7,3,5,8]$

تعریف operator: اپراتورها یا عملگرهای ما در این مساله، 4 عمل بالا پایین چپ و راست می باشند. با توجه به استیت ورودی (یعنی ماتریس ابتدایی داده شده) باید با استفاده از 4 عمل بیان شده به استیت نهایی یا goal state برسیم.

تعریف transition model: با هر یک از اعمال بالا پایین چپ راست که شرح داده شد به استیت جدیدی می رویم. مثلاً اگر استیت مثال زده شده در بالا را در نظر بگیریم 4 حالت حرکت به صورت زیر خواهند شد: بالا- $[1,0,4,6,2,7,3,5,8]$ ، پایین- $[1,2,4,6,5,7,3,0,8]$ ، چپ- $[1,2,4,0,6,7,3,5,8]$ و راست- $[1,2,4,6,7,0,3,5,8]$. استیت های نمایش داده شده معادل با ماتریس 2 بعدی 3×3 ای هستند که در صورت انتخاب هر عملگر به دست می آیند.

تعریف goal test: گل تست یعنی مقایسه هر استیت جدید به دست آمده با استیت نهایی که همان استیت مرتب شده $[1,2,3,4,5,6,7,8,0]$ است. در صورتی که معادل بودند یعنی با استفاده از سلسله اعمالی توانستیم از استیت شروع به جواب برسیم. در صورتی که برابر نبودند به پیدا کردن جواب ادامه می دهیم.

تعریف path cost: زمانی که یک عملگر انتخاب می شود و از استیتی به استیت دیگر می رویم، هزینه مسیرمان یک واحد اضافه می شود. یعنی در نهایت وقتی از استیت ابتدایی به goal state رسیدیم، جمع تعداد عملگرهای انتخاب شده برای رسیدن به استیت هدف برابر path cost یا هزینه آن مسیر است.

تابع هیوریستیک: برای این کار طبق درس دو تابع وجود داشتند به یکی از آن ها با استفاده از محاسبه فاصله Manhattan به حل جواب می پرداخت و دیگری تابع number of misplaced tiles بود که بر اساس تعداد خانه هایی که در جای خود نیستند به محاسبات می پرداخت. هر دو تابع admissible و قابل استفاده اند ولی از آن جایی که تابع اول یعنی Manhattan distance مقادیر بزرگتری تولید می کند بنابراین سریع تر به جواب می رسیم پس استفاده از آن بهتر است. (هر دو در نوتبوک آورده شده اند)

- حل مساله داده شده توسط الگوریتم ها در نوتبوک صورت گرفته است و همچنین مجموعه اعمال برای رسیدن به جواب نیز در فایل اکسل ذخیره شده است.

- در فایل excel ساخته شده جواب حاصل از الگوریتم های قرارگرفته در نوتبوک روی مثال های داده شده، ذخیره شده اند. جواب تمام مثال ها با A^* نوشته شده است و سایر الگوریتم ها بجز IDS نیز در صورتی که زمان اجرای آن ها از ۲ دقیقه بیشتر نشد، جوابشان در فایل اکسل قرار گرفت. همچنین برای الگوریتم های غیر A^* تنها با استفاده از ۳ مثال که هرکدام از بخش آسان و متوسط و سخت بودند به میانگین گرفتن پرداخته شد. چون اگر تعداد بیشتر از مثال های آسان در میانگین گرفتن استفاده می شد به میانگین کاذب و نادرست می رسیدیم.

- نتایج: پس از بررسی جواب ها در فایل اکسل می توان نتیجه گرفت که A^* با اختلاف هم سریع تر است و هم فضای کمتری اشغال می کند. سپس DFS و IDS و BFS و UCS با مقایسه مثال های محدود به ترتیب سریع تر بودند. در صورتی که تعداد مثال های بیشتری در میانگین گیری لحاظ شود می توان به نتیجه دقیق تری دست یافت. فضای اشغال شده A^* از سایرین بهتر بود و سپس UCS و IDS و BFS و DFS قرار گرفتند. (برای میانگیری دقیق تر باید از تعداد مثال بیشتری استفاده کرد. علاوه بر آن تابع محاسبه فضای مصرفی دقت کافی را ندارد) اعداد به دست آمده در فایل اکسل قرار دارند.