

گزارش پروژه اول هوش مصنوعی

کوروش روحی

در این ۳ کد از دو کلاس Card و Node برای ذخیره کردن اطلاعات هر state استفاده می‌شود.

الگوریتم BFS به این شکل استفاده شده است که ابتدا ریشه از نظر هدف بودن چک می‌شود و سپس به دو لیست frontier و explored اضافه می‌شود و تا زمانی که یا به هدف برسیم یا لیست frontier ما خالی شود، فرزندان هر node در ابتدای لیست frontier را چک می‌کنیم و سپس آن‌ها را در صورت هدف نبودن به لیست explored اضافه می‌کنیم.

در الگوریتم IDS، به طور مرتب تا یک عمق مشخص الگوریتم LDS صدا زده می‌شود که تا یک عمق مشخص چک شود برای پیدا کردن هدف.

در الگوریتم A^* ، مانند BFS عمل می‌کنیم با این تفاوت که node ها را بعد از خارج کردن از لیست frontier چک می‌کنیم که هدف هستند یا خیر

و قبل از خارج کردن اولین عضو لیست frontier آن را بر اساس تابع f مرتب می‌کنیم که در الگوریتم ما تابع هزینه عمق node است و تابع هیوریستیک ما قرینه مجموع ردیف های تک رنگ و ردیف های نزولی است.

از نظر تعداد node های تولید شده الگوریتم IDS بیشتر از دیگر الگوریتم ها است. بعد از آن الگوریتم BFS است و بعد از الگوریتم A^*

الگوریتم A^* اگر دارای یک هیوریستیک بهینه باشد بسیار می‌تواند در استفاده حافظه و کوتاه کردن زمان رسیدن به هدف مفید باشد.

ولی امکانش هست در الگوریتم IDS و A^* با اینکه به جواب برسیم و حتی زودتر از BFS برسیم ولی در عمق بیشتری به جواب برسیم.