محمدرضا کیانی‌فر
۸۱۰۱۹۴۳۸۸

نحوه‌ی مدل کردن مسئله

در BFS و IDS:

در این دو الگوریتم، استیت‌های من، چیدمان وزیرها در صفحه می‌باشند. استیت‌های همسایه در جایگاه یک وزیر با هم تفاوت دارند و این تفاوت نیز، یک حرکت شاهی (یعنی ۸ حرکت ممکن برای جابه‌جایی یک شاه) است. همچنین ممکن است تفاوتی نیز نداشته باشند و هیچ وزیری در استیت بعدی حرکت نکرده باشد. حال من یک درخت با ارتفاع ۹ در نظر می‌گیرم. در سطح اول، حالت اولیه‌ی مسئله یعنی ۸ مکان اولیه‌ی ۸ وزیر که از ورودی دریافت می‌شود، قرار دارد. سپس در سطح دوم، جابه‌جایی‌های ممکن برای حرکت شاهی وزیر اول + ساکن بودن آن قرار می‌گیرند؛ در واقع ممکن است که در سطح دوم نهایت ۹ حالت ایجاد شود. (بعضی از این حالت‌ها به علت خارج شدن وزیر از صفحه، حالت‌های مجازی نیستند و من آن‌ها را در نظر نمی‌گیرم.) سپس برای سطح سوم، حرکت شاهی وزیر دوم + ساکن بودن حالت‌های جدیدی که در سطح دوم ایجاد شده اند را بررسی می‌شوند و حالت‌های مجاز در سطح سوم قرار می‌گیرند. این درخت تا سطح ۹ ام می‌تواند ادامه پیدا کند و سپس در سطح ۹، جابه‌جایی های ممکن برای وزیر ۸ ام نیز تمام می‌شود و ماکل فضای حالت حرکت‌های تکی وزیرها به اطرافشان را بررسی کرده‌ایم.

هدف در این پیمایش این درخت، رسیدن به استیتی است که در آن وزیرها همدیگر را تهدید نکنند و اگر این کار با حرکت شاهی ممکن باشد تا سطح ۹ ام به آن می‌رسیم. در صورتی که به استیت هدف نرسیدیم(که برای ۳ ورودی این پروژه، به جواب می‌رسیم)، می‌توانیم حرکت‌ها را زیادتر کنیم. مثلا به جای بررسی ۹ حالت برای یک وزیر (Expansion با مقدار ۹)، ۸ حالت بیشتر برای جابه‌جایی دوتایی یک وزیر نیز در نظر بگیریم. (Expansion با مقدار ۱۷) تعداد حرکت‌ها تا حد ۵۷ حرکت قابل زیاد شدن است. (۸*۷ حداکثر تعداد حالت‌های ممکن برای حرکت یک وزیر و +۱ برای حرکت ندادن وزیر) با بررسی مسئله با اکسپنشن ۵۷، در صورتی که جوابی با یک بار حرکت هر وزیر موجود باشد خواهیم یافت.

در A*:

در این الگوریتم، استیت‌های من همان چیدمان وزیرها در صفحه است. در صورتی که دو استیت به هم یال داشته باشند، قطعا جایگاه یک وزیر در آن‌ها متفاوت است. (حرکت شاهی انجام شده) هدف رسیدن به استیتی است که در آن وزیرها همدیگر را تهدید نکنند. هر استیت می‌تواند به حداکثر ۶۴ استیت دیگر وصل شود که حداکثر حالت‌های ناشی از حرکت‌های شاهی هر یک از ۸ وزیر آن استیت است.

توضیح الگوریتم‌ها

BFS:

در این الگوریتم، من از حالت اولیه مسئله که در سطح اول قرار دارد، شروع می‌کنم و با بررسی حرکت‌های شاهی وزیر اول، تمام حالت‌های سطح دوم را می‌سازم. در ضمن ساخته شدن حالت جدید نیز، رسیدن به هدف را چک می‌کنم. سپس برای هر یک از حالت‌های سطح دوم، تمام جابه‌جایی‌های مجاز وزیر دوم را به سطح سومم اضافه می‌کنم. این عملیات تا سطح ۹ ام می‌تواند ادامه پیدا کند. نمونه‌ی دوم و سوم در سطح ۸ ام به پایان می‌رسند و جواب پیدا می‌شود و ورودی اول در سطح نهم، لذا اختلاف زمان قابل توجهی بین آن‌ها وجود دارد.

IDS:

در این الگوریتم من ابتدا در یک متغیر حداکثر ارتفاع برای پیمایش را تعیین می‌کنم و آن را برابر ۱ قرار می‌دهم. از حالت اولیه‌ی مسئله که در سطح اول قرار دارد، شروع می‌کنم. در صورتی که از حداکثر ارتفاع پیمایش نمی‌گذشتم، وزیر اول را یک حرکت شاهی می‌دهم. سپس به سطح دوم می‌روم و با بررسی حداکثر ارتفاع پیمایش، وزیر دوم را تغییری می‌دهم. این عملیات را تا سطح نهم تکرار می‌کنم و در این حین رسیدن به حالت هدف را بررسی می‌کنم. اگر تا انتهای مسیر (سطح ۹ ام) رفتم و به جواب نرسیدم، به یک سطح بالاتر باز می‌گردم و به حالت جدیدی می‌روم. در صورتی که از حداکثر سطح مجاز نیز می‌گذشتم، پس از پایان بررسی آن سطح، الگوریتم را از ابتدا و با یک واحد افزایش متغیر حداکثر ارتفاع تکرار می‌کنم.

A*:

در این الگوریتم، من دو مجموعه استیت دارم: استیت‌های پیمایش شده و استیت‌های پیشگام. ابتدا حالت اولیه را در مجموعه‌ی استیت‌های پیمایش شده قرار می‌دهم و برای آن، تمام حالت‌های مجاز حرکت شاهی هر یک از وزیرها را بررسی می‌کنم (در مجموع ۶۴ حرکت بررسی می‌شوند) و آن‌ها به همراه مقدار هیوریستیک و تابع هزینه‌ی مسیر را ذخیره می‌کنم. سپس در هر مرحله، از بین مجموعه حالت‌های پیشگام، حالت با هزینه‌ی مینیموم را به دست آورده، آن را از مجموعه‌ی پیشگام حذف و به مجموعه‌ی پیمایش شده اضافه می‌کنم. سپس ۶۴ حالت آن را بررسی کرده و در صورت صحت و وجود نداشتن در حالت‌های پیشگام، آن‌ها را به حالت‌های پیشگام اضافه می‌کنم. این کار را تا رسیدن به حالت هدف تکرار می‌کنم.

توضیح تابع هیوریستیک

تابع هیوریستیک من برای الگوریتم A*، ۲ برابر تعداد تهدیدهای وزیرها در یک حالت است.(h(x))

تابع هزینه‌ی مسیر من نیز، تعداد حرکت‌های شاهی وزیرهای یک حالت نسبت به حالت اولیه است. (g(x))

حال تابع هزینه‌ی کل ما یعنی f(x) = g(x) + h(x)، تابع خوبی است چرا که با دوبرابر شدن تعداد تهدیدها، اهمیت آن در هزینه نسبت به جابه‌جایی وزیرها لحاظ می‌شود، یعنی حالتی که یک حرکت اضافه موجب یک عدد کاهش تعداد تهدیدهاست، حالت بهتری نسبت به انجام حرکت کمتر و تهدید بیشتر است و موجب می‌شود به سرعت به جواب برسیم.

تفاوت‌های الگوریتم‌ها

از آن‌جایی که الگوریتم BFS و IDS به شکل ناآگاهانه است، به مراتب زمان بسیار بیشتری نسبت به الگوریتم A* طول می‌کشند. چرا که بدون هیچ سیاست و هوشمندی، کل فضا را جست‌وجو می‌کنند اما A* با یک روشی هوشمندانه‌تر در جست‌وجو خواهد بود. هزینه‌ی زمانی الگوریتم‌های BFS و IDS از جنس O(b^d) است که در آن b حداکثر تعداد شاخه‌ها و m طول بیشترین مسیر ممکن در گراف است اما در A* هزینه‌ی زمانی الگوریتم برابر است با تعداد نودهایی که در آن‌ها g(n) + h(n) از جواب اصلی کمتر مساوی است. BFS نیاز به حافظه‌ی بیشتری نسبت به IDS دارد چرا که هزینه‌ی حافظه‌ای BFS از جنس O(b^d) است در حالی که هزینه‌ی حافظه‌ای O(b*d) است. هزینه‌ی حافظه‌ای A* نیز مانند هزینه‌ی زمانی الگوریتم آن است.

شباهت‌های الگوریتم‌ها

هر سه الگوریتم Complete هستند. BFS و IDS در صورتی که هزینه‌ی قدم‌ها یکسان باشند، Optimal هستند و A* نیز در صورتی که الگوریتم هیوریستیک قابل قبول یا Admissable باشد، Optiaml است.

زمان اجرای الگوریتم

	sample 1	sample 2	sample 3	Average
BFS	113.4	34.5	38.6	62.14
IDS	73.3	34.0	40.1	49.13
A*	0.5	0.8	2.1	1.13
	62.4	23.1	36.9	

تعداد حرکت‌های انجام شده

	sample 1		sample 2		sample 3	
	Steps	Moves	Steps	Moves	Steps	Moves
BFS	۲۰۹۴۵۱۰	۶	۱۱۲۷۱۷۷	۷	۱۰۷۸۰۹۸	۷
IDS	۲۴۵۴۱۸۳	۶	۱۲۸۸۷۰۹	۷	۱۲۵۵۳۲۲	۷
A*	۲۱۹	۶	۳۳۷	۷	۶۸۱	۷

زمان اجرای الگوریتم برای ورودی‌های جدید

	sample a	sample b	sample c	Average
BFS	2.36	19.83	52.6	24.93
IDS	2.54	18.68	40.1	20.44
A*	0.01	0.1	0.2	0.10
	1.63	12.87	30.9	

تعداد حرکت‌های انجام شده برای ورودی‌های جدید

	sample a		sample b		sample c	
	Steps	Moves	Steps	Moves	Steps	Moves
BFS	۸۶۴۱۲	۳	۶۰۶۹۴۶	۴	۱۴۰۷۰۰۰	۵
IDS	۹۷۲۷۳	۳	۷۰۴۲۱۹	۴	۷۰۴۲۸۳	۵
A*	۶	۳	۳۷	۴	۶۹	۵