به نام خدا

تمرین تئوری سری اول هوش مصنوعی

پوریا صفائی ۹۸۱۱۰۴۰۲

سوال ۱)

	سیستم تشخیص صدا	پهپاد
Partially Observable/Fully Observable	Fully	Partially
Deterministic/Stochastic	Deterministic	Stochastic
Episodic/Sequential	Episodic	Sequential
Static/Dynamic	Static	Dynamic
Single Agent/Multi Agent	Single Agent	Multi Agent

سوال ۲)

الف)

این ادعا را با استقرا روی تعداد جدول ها اثبات میکنیم. برای m=1 حکم برقرار است و کافیست خانه خارج شده را به جدول بازگردانیم. حال برای m + 1 حکم را اثبات میکنیم. رنگی که خارج از جداول است را k در نظر بگیرید. حال جدول j ≠ k را در نظر بگیرید. ابتدا این جدول را تماما به رنگ j در می‌آوریم. اگر در مرحله ای از این الگوریتم جدول تماما به رنگ j در آمد، الگوریتم متوقف شده و چون رنگی که خارج است j نیست، بنابراین خانه خالی در جدولی غیر از j وجود دارد. پس حالا m جدول با m رنگ(غیر از j که همه خانه های به این رنگ در جدول j ام هستند) وجود دارند که یک خانه از آن برداشته شده. پس طبق فرض استقرا میتوانیم با دنباله ای از حرکات به هدف مسئله برسیم.

حال الگوریتمی که جدول j را تماما به رنگ j در می‌آورد به این صورت است: در هر مرحله، خانه خالی اگر در جدولی غیر از j است، جای خانه خالی را با یکی از خانه های j که به رنگ j نمی‌باشد عوض میکنیم. حال که خانه خالی در جدول j است، جای خانه خالی را با خانه ای در خارج از j که به رنگ j میباشد عوض میکنیم و همین الگوریتم را ادامه میدهیم تا جدول j تماما به رنگ j در آید. و سپس با فرض استقرا دنباله ای از حرکات یافت خواهد شد که ما را به هدف مسئله برساند.

ب)

به هر جدول، یک عدد نسبت میدهیم و آن عدد ماکزیمم تعداد خانه ها با یک رنگ است و آن را با c_j نمایش میدهیم. حال برای جدول j، h_j را به صورت $h_j = m^2 - c_j$ تعریف میکنیم و هیورستیک مورد نظر برای هر استیت از بازی به صورت زیر است:

$$H(s) = \sum_{j=0}^{m} h_j$$

با توجه به تعریف هیورستیک، میتوان مشاهده کرد که فاصله تا هدف را آندراستیمیت میکند. چرا که برای هر جدول باید حداقل به اندازه خانه هایی که به رنگ آن نیست(معادلا حاکثر به اندازه همان h_j) مرحله جابه جایی صورت بپذیرد تا همه یک رنگ شود. از طرفی هر جابه جایی بین ۲ جدول متفاوت مقدار h_j یکی از جدول ها را حداکثر یک واحد کاهش میدهد(چون یک خانه از یک جدول به جدول دیگر منتقل میشود و میتواند مقدار h را حداکثر برای یکی از آنها به ازای ذنگ خانه جابه جا شده یک واحد تغییر دهد). پس حداقل به اندازه هیورستیک تعریف شده تا هدف فاصله داریم. پس این هیورستیک

دارای خاصیت admissible است. همین شرایط برای هیورستیک مسئله، شرط consistency را هم برقرار میکند. رابطه زیر برای این شرط باید برقرار باشد:

$$H(b) - H(a) \leq cost(a, b)$$

و این رابطه به این دلیل برقرار است که هزینه رسیدن از a به b، حداقل به اندازه جابه جایی خانه بین جدول های متفاوت در استیت های a و b است و ممکن است تعدادی جابه جایی هم داخل هر جدول لازم باشد که آن هیورستیک را عوض نمیکند. از طرفی جابه جایی بین جدول ها، طبق نکته ای که بیان شد، هیورستیک را حداقل یک واحد تغییر میدهد. پس داریم:

$$H(a) + cost(a, b) \geq H(b)$$

که معادل همان رابطه بالا است.

سوال ۳)

الف) مسیر DFS:

$$a \to b \to c \to e \to f \to g \to d$$

مسیر BFS:

$$a \to b \to c \to d \to e \to f \to g$$

ب) مراحل طی شده:

$$a(0) \to b(1) \to c(2) \to d(4) \to e(5) \to f(8) \to g(17)$$

مسیر از مبدا به مقصد:

$$a \to c \to e \to f \to g$$

ج)

تابع H_2 نمیتواند تابع معتبر باشد چرا که $H_2(b) = 19$ اما هزینه واقعی مسیر b,c,e,f,g برابر با ۱۹ است. پس مقدار آن بیشتر از مقدار واقعی است . این تابع نمیتواند معتبر باشد.

اما H_1 این خاصیت را دارد چرا که کمترین فاصله هر راس تا راس g، بزرگتر مساوی مقدار تابع H_1 برای آن راس میباشد.

د)

در الگوریتم حریصانه، در بین کاندیدها راسی انتخاب میشود که مقدار هیورستیک آن کمترین است. پس مراحل در این حالت برای H_1 برابر میشود با:

$$a(0) \to b(H_1 = 3) \to d(H_1 = 10) \to g(H_1 = 0)$$

و مسیر از مبدا به مقصد به صورت زیر میباشد:

$$a \to d \to g$$

ه)

در الگوریتم *A، در بین کاندیدها راسی انتخاب میشود که مقدار هیورستیک بعلاوه هزینه واقعی پرداخت شده تا آن کمترین است. پس مراحل در این حالت برای H_2 برابر میشود با:

$$a(0) \to c(H_2 + c = 16) \to e(H_2 = 15) \to d(H_2 = 16) \to f(H_2 = 16) \to g(H_2 = 17\ goal)$$

و مسیر از مبدا به مقصد برابر با:

$$a \to c \to e \to f \to g$$

سوال ۴)

الف)

استیت ها: تمام جایگشت های ممکن از اعدات ۱ تا n

کنش ها: در هر مرحله میتوانیم اندیس i بین ۱ تا n را انتخاب کنیم و به حالتی برویم که i خانه اول جایگشت قرینه حالت فعلی است. دقت کنید که در حالتی که i = 1 است جایگشت تغییری نمیکند.

ضریب انشعاب: در تمام حالات به جز i = 1 به حالت متمایز دیگری میرویم. پس در این مسئله ضریب انشعاب n – 1 میباشد.

حالت اولیه: رشته ورودی است که جایگشتی دلخواه از ۱ تا n میباشد.

حالت نهایی: دنباله مرتب شده از عدد ۱ تا عدد n

ب)

همانطور که در بخش قبل گفتیم فضای حالت n! میباشد و طبق تخمین استرلینگ برای فاکتوریل این مقدار برابر با O(n^n) میباشد.

ج)

ابتدا دقت کنید که هر عملی در هر حالتی تعداد نقاط گسست را نهایتا میتواند یک واحد تغییر میکند چرا که اگر i خانه اول را جابه جا کنیم، تعداد نقاط گسست بین i خانه اول و n-i خانه دوم ثابت میماند و تنها ممکن است که خانه iام و i+1ام گسست بینشان تغییر کند. با توجه به این نکته admissible بودن این هیوریستیک به دست می آید چرا که در حالت هدف، تعداد گسست ها صفر است و با توجه به نکته هر عمل حداکثر یک واحد گسست را کاهش میدهد. پس هیورستیک مورد نظر کران پایینی از فاصله واقعی است. از طرفی میتوان شرط consistency را هم بررسی کرد. هزینه این که از یک استیت به استیت دیگری برویم، حداقل به اندازه اختلاف بین تعداد گسست های آنها است. چرا که هر عمل نهایتا یک واحد گسست را تغییر میدهد. پس داریم:

$$|h(b) - h(a)| \leq cost(a,b) \rightarrow h(b) \leq cost(a,b) + h(a)$$

پس شرط کانسیستنسی نیز برقرار است.

سوال ۵)

الف) صحیح است چرا که اگر مقدار k برابر با ۱ باشد آنگاه از بین همسایه های گره بهترین گزینه را انتخاب میکند که همان استراتژی هیل کلایمینگ است.

ب) صحیح است چرا که اگر مقدار دما را صفر قرار دهیم، مقدار نمای تابع به منفی بینهایت و در نتیجه مقدار خود تابع به صفر میل میکند. پس احتمال اینکه یک همسایه که مقدار ارزش آن از وضعیت فعلی کمتر است انتخاب شود صفر است. بنابراین تا زمانی ادامه میدهیم تا به همسایه ای برسیم که وضعیت

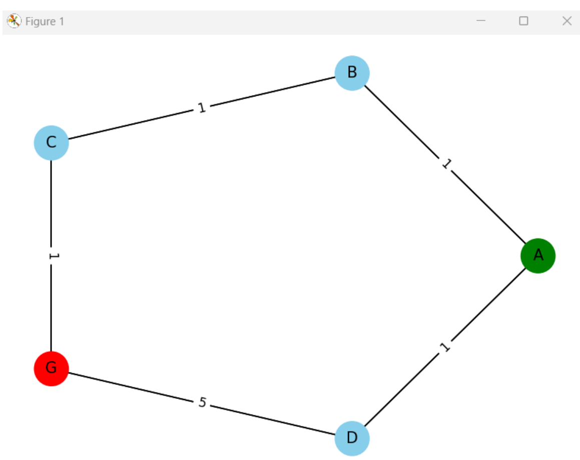
بهتری از استیت فعلی دارد که این همان عملکرد الگوریتم first choice hill climbing است. در این حالت میتوان به سادگی دید که این هیورستیک شرط ad,issi

ج) لزوما به این صورت نیست. چرا که الگوریتم ژنتیک هر استیت را با احتمال یکنواختی جهش میدهد و در نهایت از نتایج به دست آمده، با توجه به نتیجه با توزیعی که متناسب با مقدار ارزش است انتخاب میکند. اما دربیم سرچ تصادفی، در بین تمام همسایه های استیت این توزیع صورت میگیرد نه فقط استیت هایی که به صورت جهش تصادفی به آنها دسترسی پیدا میکنیم.

د) این گزاره صحیح است چرا که این الگوریتم بدون کراساور و سلکشن، تنها ژن ها را به صورت تصادفی و توزیع یکنواخت جهش میدهد. هر جهش تصادفی معادل این است که از یک استیت به استیت مجاور به صورت تصادفی انتقال پیدا کنیم که معادل همان رندوم واک میباشد.

ه)

برای این بخش میتوانیم یک مثال نقض ارائه کنیم:



برای این گراف مسیر بازگشتی الگوریتم Uniform Cost Search برابر با ABCG است که وزن ۳ را دارد. اما اگر به طور مثال مقدار ۱۰۰ را با وزن تمام یال ها جمع کنیم، مسیر بازگشت آنگاه کوتاهترین مسیر برابر ADG میشود.

و)

این گزاره صحیح است. اثبات:

$$\begin{cases} h(a) + c(a,b) \geq h(b) \\ g(a) + c(a,b) \geq g(b) \end{cases} \rightarrow h(a) + g(a) + 2c(a,b) \geq h(b) + g(b)$$

$$k(s) = \frac{h(s) + g(s)}{2} \rightarrow 2k(s) + 2c(a,b) \geq 2k(b) \rightarrow k(s) + c(a,b) \geq k(b)$$

که این نتیجه نشان میدهد تابع k(s) که میانگین h و g است نیز شرط consistency را دارا میباشد.