باسمه تعالی

تکلیف سری اول درس هوش مصنوعی

سارا برادران (شماره دانشجویی : ۹۶۲۴۱۹۳)

سوال ۱)

$x1$=65413532 , $x2$=87126601 , $x3$=23921285 , $x4$=41852094

الف)

F(x) = $(a+b)-(c+d)+(e+f)-(g+h)$

F(x1) = (6+5)–(4+1)+(3+5)–(3+2) = 9

F(x2) = (8+7)–(1+2)+(6+6)–(0+1) = 23

F(x3) = (2+3)–(9+2)+(1+2)–(8+5) = -16

F(x4) = (4+1)–(8+5)+(2+0)–(9+4) = -19

F(x2) > F(x1) > F(x3) > F(x4)

برازندگی کل جمعیت = (19-)+(16-)+9+23 = 3-

ب)

Single point crossover

$x2$=871 | 26601

$x1$=654 | 13532

Child 1 = x5 = 87113532 → F(x5) = (8+7)–(1+1)+(3+5)–(3+2) = 16

Child 2 = x6 = 65426601 → F(x6) = (6+5)–(4+2)+(6+6)–(0+1) = 16

توضیحات: در صورت سوال ذکر شده است دومین و سومین فرد با توجه به قسمتهای قبل **دومین و سومین فرد از نظر برازندگی** را برای عملیات crossover استفاده کرده ام یعنی دو عضو x1, x3 .

Two point crossover

$x1$=65 | 4135 | 32

$x3$=23 | 9212 | 85

Child 1 = x7 = 65921232 → F(x7) = (6+5)–(9+2)+(1+2)–(3+2) = -2

Child 2 = x8 = 23413585 → F(x8) = (2+3)–(4+1)+(3+5)–(8+5) = -5

توضیحات: در صورت سوال ذکر شده است اولین و سومین فرد با توجه به قسمتهای قبل **اولین و سومین فرد از نظر برازندگی** را برای عملیات crossover استفاده کرده ام یعنی دو عضو x2, x3 .

Uniform crossover

x2=87126601

x3=23921285

Child 1 = x9 = 83926201 → F(x9) = (8+3)–(9+2)+(6+2)–(0+1) = 7

Child 2 = x10 = 27121685 → F(x10) = (2+7)–(1+2)+(1+6)–(8+5) = 0

پ) X2 , X6 , X5, X1 افراد با بیشترین میزان fitness در جمعیت هستند پس این افراد را برای جمعیت نسل بعد انتخاب خواهیم کرد و x3 , x4 , x7 , x8 , x9 , x10 از جمعیت نسل بعد حذف خواهند شد.

برازندگی کل جمعیت جدید = 9+23+16+16 = 64

برازندگی کل جمعیت جدید به نسبت جمعیت اولیه رشد چشمگیری داشته و از 3- به 64 تغییر کرده است.

ت) برای یافتن جواب بهینه کافی است تابع برازندگی را بیشینه کنیم :

تابع برازندگی زمانی بیشینه می شود که ترم های منفی کننده کمترین مقدار و ترم های مثبت کننده بیشترین مقدار را داشته باشند به عبارت دیگر اگر مقادیر زیر را به ژن ها اختصاص دهیم این بیشینه حاصل می گردد.

a = b = e = f = 9

c = d = g = h = 0

X_m = 99009900 → F(X_m) = (9+9)–(0+0)+(9+9)–(0+0) = 36

ث) عملگر جهش را می توان اینگونه در نظر گرفت که با یک احتمال کم اما ناصفر p یکی از ژن های کروموزوم مقدار فعلی خود را از دست داده و یک مقدار تصادفی بین ۰ تا ۹ برای آن اتخاذ شود.

ج) خیر بدون عملگر جهش امکان رسیدن به بهینه سراسری با توجه به مقادیر جمعیت اولیه امکان پذیر نمی باشد زیرا یکی از شرایط لازم برای دست یافتن به بهینه سراسری به کمک single point crossover این است که در حداقل یک عضو جمعیت ژن اول یعنی a مقدار ۹ داشته باشند درحالی که در جمعیت اولیه هیچ عضوی دارای این ویژگی نمی باشد.

سوال ۲)

الف) الگوریتم همسایه های بدتر کننده را انتخاب نخواهد کرد و ضمن رسیدن به اولین همسایه بهبود دهنده آن را به عنوان استیت گام بعد انتخاب می کند پس مشابه first choice hill climbing خواهد بود.

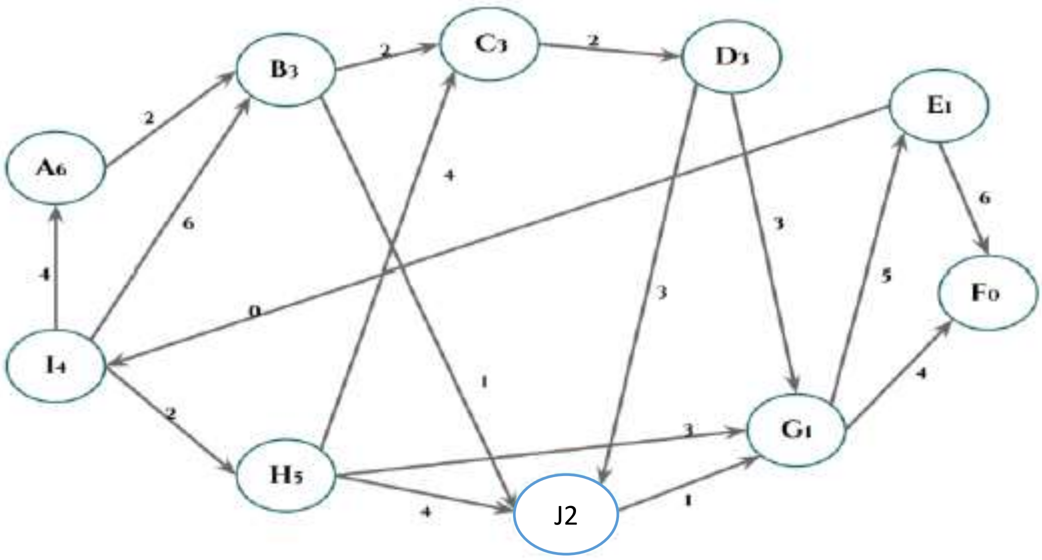
ب) الگوریتم کاملا تصادفی و به نوعی random walk عمل خواهد کرد.

پ) الگوریتم مانند این است که یک استیت موجود بوده و در هر مرحله همسایه های آن استیت را تشکیل داده و بهترین را به عنوان استیت مرحله بعد انتخاب کنیم پس مشابه steepest ascent hill climbing خواهد بود.

ت) نوعی brute force اتفاق می افتد که کلیه فضای حالت را مورد بررسی قرار می دهد.

ث) الگوریتم با توجه به اینکه تنها یک عضو جمعیت دارد لذا اگر عضو با خودش ترکیب شود و فرزندی تشکیل دهد همچنان همان والد اولیه به عنوان فرزند شکل خواهد گرفت پس از عمل crossover عضو جدیدی تولید نخواهد شد لذا بسته به اینکه الگوریتم ژنتیک از چه تابعی برای mutation استفاده می کند در هر گام صرفا با عمل mutation بر روی تنها عضو جمعیت یک همسایه ایجاد خواهد شد که از میان ۲ عضو موجود بهترین آن را به عنوان نسل بعد انتخاب میکنیم پس به نوعی مشابه hill climbing عادی خواهد بود.

سوال ۳)



الف) اگر هدف یافتن کمترین هزینه باشد آنگاه مسیر زیر انتخاب خواهد شد:

A → B → J → G → F → COST = 6+3+2+1+0 = 12

اگر هدف یافتن بیشترین پاداش باشد آنگاه یکی از دو مسیر زیر انتخاب خواهد شد:

A → B → C → D → G → E → F → REWARD = 2+2+2+3+5+6 = 20

A → B → C → D → J → G → E → F → REWARD = 2+2+2+3+1+5+6 = 21

ب) مسیری که الگوریتم Tabu search در مثال فوق برای کمترین هزینه و بیشترین پاداش می یابد دقیقا مشابه مسیر هایی است که الگوریتم hill climbing در قسمت قبلی یافته بود چرا که اساسا Tabu search دارای یک حافظه است که موجب می شود اگر در یک گام به استیت x رفتیم تا حداقل tt گام بعد نتوانیم مجددا به x برویم به علاوه Tabu search در هر گام بهترین همسایه را انتخاب می کند چه بهتر از حالت فعلی باشد و چه بدتر. از آنجایی که در مثال بالا توسط الگوریتم hill climbing همواره بهترین همسایه انتخاب شده بود و در طول مسیر های بدست آمده هیچ استیتی دوبار مشاهده نشده بود لذا الگوریتم Tabu search نیز دقیقا به همان مسیر های مرحله قبل خواهد رسید.

پ) الگوریتم ها در یافتن کمترین هزینه در بهینه سراسری متوقف می شوند. اما همین الگوریتم ها در یافتن بیشترین پاداش در بهینه محلی متوقف می شود زیرا مسیری موجود است که نسبت به جواب بدست آمده پاداش بیشتری می دهد مانند مسیر زیر :

A → B → C → D → J → G → E → I → H → G → F

→ REWARD = 2+2+2+3+1+5+0+2+3+4 = 24

ت) اگر منظور یافتن مسیری است که نسبت به پاداشش کمترین هزینه را داشته باشد می توانیم یک تابع هدف به صورت :

$$F = Reward / cost$$

تعریف و سپس الگوریتم hill climbing را با هدف بیشینه کردن این تابع اجرا کنیم. به عبارتی برای هر همسایه مقدار F را حساب کرده و سپس از میان همسایه ها بهترین را انتخاب نماییم.

در غیر اینصورت در گراف فوق مسیری موجود نمی باشد که به طور مطق از میان همه مسیرهای ممکن هم کمترین هزینه و هم بیشترین پاداش را داشته باشد زیرا طبق نتایج قسمت های قبل مسیری که کمترین هزینه را می دهد جدای از مسیر با بیشترین پاداش است.

سوال ۴)

الف) در این سوال می توان اعضای فضای حالت را تمام نقاط با مختصات (X,Y) موجود در صفحه در نظر گرفت. نمایش یک عضو نیز به همین شکل یعنی (X,Y) خواهد بود.

تعریف همسایگی : همسایه های یک نقطه با مختصات (X,Y) را نقاط مجاور آن یعنی نقاط با مختصات *(X+1,Y), (X,Y+1), (X,Y-1), (X-1,Y), (X+1,Y+1), (X-1,Y-1), (X+1,Y-1), (X-1,Y+1)* در نظر می گیریم. در حالت کلی می توان گفت هر عضو از فضای حالت دارای ۸ همسایگی می باشد.

تابع هدف : تابع هدف را برای هر عضو می توان فاصله میان آن عضو تا نقطه G در نظر گرفت به عبارتی برای عضوی از فضای حالت با نمایش (X, Y) مقدار تابع هدف به شکل زیر قابل محاسبه است:

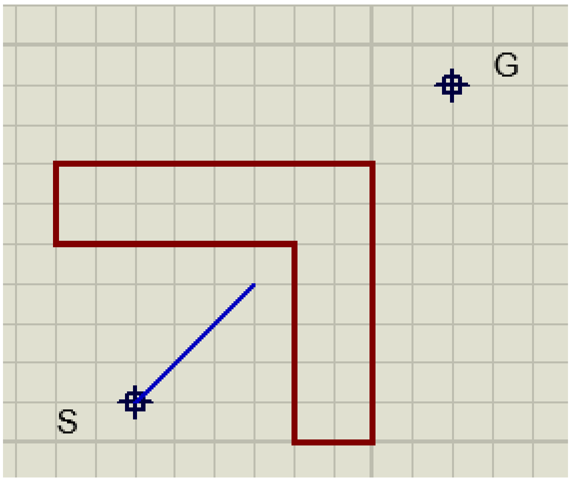
$$k + \sqrt{(X_G - X)^2 + (Y_G - Y)^2}$$

در فرمول فوق k میزان جریمه ای است که در صورت موجودیت نقطه در داخل و یا روی خطوط موانع مقداری بزرگ مثلا مثبت بی نهایت اخذ کرده و در غیر اینصورت مقدار صفر اخذ خواهد کرد.

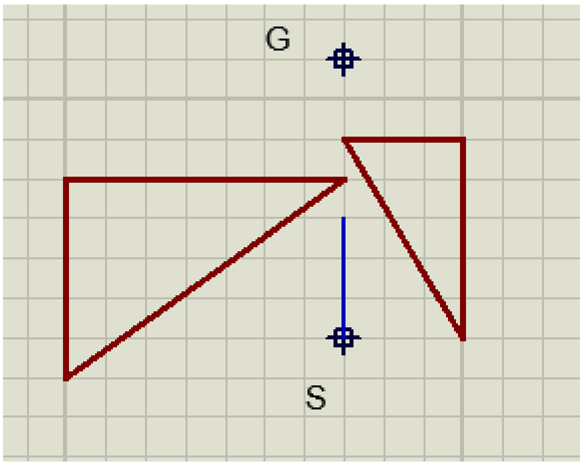
حال می توان گفت نحوه کار الگوریتم تپه نوردی به این صورت است که از عضو آغازین S شروع به کار کرده و در هر استیت همسایه های آن را تشکیل داده سپس همسایه ای که مقدار تابع هدف برای آن کمینه است را انتخاب کرده و در مرحله بعد به آن می رود. به علاوه در صورتی که هیچ همسایه بهبود دهنده ای موجود نباشد آنگاه الگوریتم متوقف

خواهد شد. در این مسئله قصد کمینه سازی تابع هدف است و در صورتی که به نقطه ای برسیم که مقدار تابع هدف برای آن صفر باشد بدین معناست که به بهینه سراسری دست یافته ایم و دنباله حاصل از نقاط طی شده همان مسیر بهینه را می دهد در غیر اینصورت احتمالا الگوریتم در یک بهینه محلی متوقف شده است.

ب) اگر تصویر زیر را به عنوان یک نمونه مسئله در نظر بگیریم آن گاه الگوریتم Hill Climbing مسیر آبی رنگ را انتخاب خواهد کرد ولی نهایتا در یک بهینه محلی متوقف خواهد شد زیرا در نقطه ای قرار گرفته که هیچ همسایه بهبود دهنده ای موجود نمی باشد.



پ) بله مثال زیر این امکان را نشان می دهد:



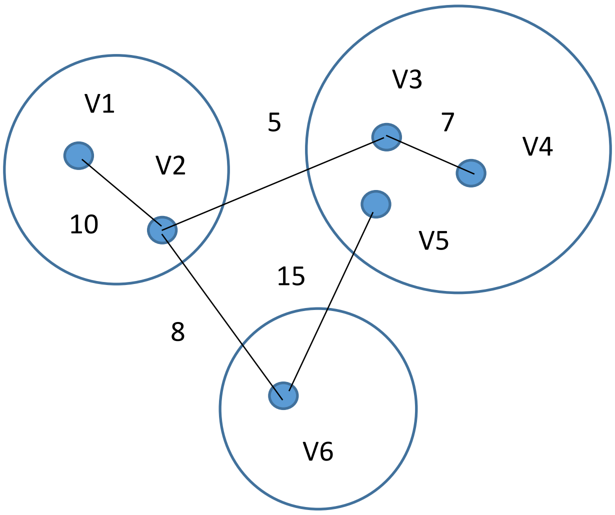
ت) خیر زیرا با کاهش دما حالت انتخاب تصادفی کاهش یافته و نهایتا به یک نقطه همگرا خواهیم شد از آن جایی که بهینه محلی در هر نقطه از landscape می تواند موجود باشد لذا احتمال همگرا شدن به بهینه محلی همواره موجود خواهد بود و تنها احتمال این امر با استفاده از الگوریتم SA کاهش می یابد اما صفر نخواهد شد. علی الخصوص اگر در دماهای پایین در محدوده بهینه محلی قرار گیریم آن گاه احتمال گریز از آن کم خواهد بود.

سوال ۵)

مسئله خوشه بندی گراف

الف) یک نمایش مناسب برای اعضای فضای جستجو را می توان به صورت دنباله های n تایی در نظر گرفت (n تعداد رئوس گراف است) به طوری که هر یک از n عضو دنباله یک عدد بین ۱ تا k باشد که این مقدار نمایانگر شماره خوشه ای است که راس متناظر در آن قرار گرفته است.

فرضا اگر شکل زیر یک عضو از search space مسئله گراف تعیین شده با k = 3 باشد نمایش این عضو به شکل زیر خواهد بود.



1	1	2	2	2	3
V1	V2	V3	V4	V5	V6

ب) تعداد کل اعضای فضای جستجو را می توان برابر $k!\binom{n}{k}k^{(n-k)}$ دانست. (در هر خوشه لزوما یک راس باید قرار داشته باشد پس انتخاب k از n را خواهیم داشت که به k! حالت میتوان یک راس را به هر خوشه نظیر کرد سپس تعداد (n-k) راس باقی مانده را می توان در هر خوشه دلخواه ۱ تا k قرار داد)

پ) چندین مدل همسایگی برای این مسئله مفروش است که در ادامه به بررسی آن ها خواهیم پرداخت:

۱- همسایگی را می توان به صورت 2-swap در نظر گرفت (بهتر است یک شرط روی 2-swap قرار دهیم و آن این است که تنها ۲ راسی که در یک خوشه قرار ندارند یعنی مقدار آن ها در دنباله نمایش عضو یکسان نیست بتوانند باهم جا به جا شوند) ایراد: این نوع تعریف همسایگی می تواند موجب جلوگیری از رسیدن به بهینه سراسری شود زیرا امکان تغییر تعداد رئوس موجود در هر خوشه امکان پذیر نیست و تنها رئوس خوشه های موجود در عضو اولیه با یکدیگر جابه جا میشوند.

۲- همسایگی را می توان اینگونه در نظر گرفت که در یک همسایه تمام رئوس در عضو فعلی ثابت و در خوشه خود باقی بمانند مگر یک راس که از یک خوشه به k-1 خوشه دیگر منتقل می شود. واضح است که این تعریف همسایگی ایراد تعریف قبلی را ندارد و هر راس می توان به صورت آزادانه میان خوشه ها جابه جا شود. البته نکته حائز اهمیت در این تعریف این است که اگر انتقال یک راس از یک خوشه منجر به خالی شدن آن خوشه گردد می بایست از آن همسایه صرف نظر کنیم.

ت) اگر همسایگی را به صورت ذکر شده (تعریف دوم) در نظر بگیریم به ازای هر عضو از فضای جستجو حداکثر تعداد n(k-1) همسایگی موجود است.(با فرض آن که انتقال هیچ راسی به خوشه دیگر برای تشکیل همسایگی نظیر آن منجر به خالی شدن یک خوشه نگردد) به علاوه حداقل تعداد همسایه ها می تواند صفر باشد در صورتی که در یک مسئله n=k باشد آنگاه انتقال هر راس منجر به خالی شدن یک خوشه می گردد پس هیچ همسایگی ای نمی توان ساخت ، اگرچه پاسخ مسئله در این مثال بدیهی و برابر : (مجموع وزن همه یال های گراف) -

است زیرا هیچ ۲ راسی در یک خوشه قرار نخواهند گرفت پس w1 = 0 و (مجموع وزن همه یال های گراف) = w2

مثلا در مثال شکل فوق به ازی هر عضو حداکثر تعداد ۶*۳ = ۱۸ همسایگی موجود است.

ث) خیر برای حل دقیق این مسئله الگوریتم چند جمله ای وجود نداشته و یک مسئله NP-hard به ازای k >= 2 می باشد. صرفا یک الگوریتم چندجمله ای غیر دقیق (2k-1)approximation برای آن موجود است.