

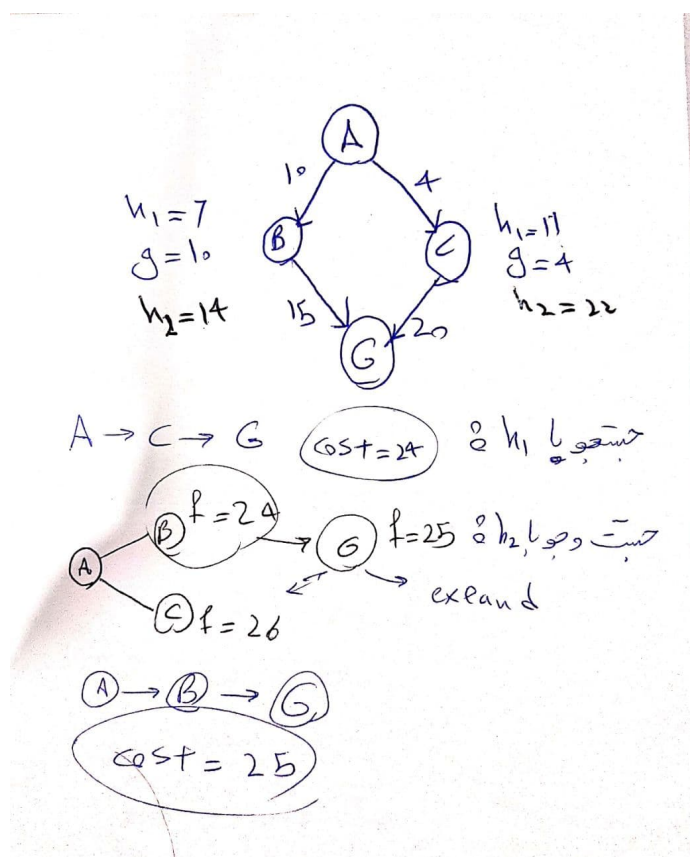
## تمرین تئوری ۲.۱ هوش مصنوعی استاد رهبان

امیر حسین باقری ۹۸۱۰۵۶۲۱

### سوالات جست و جو

آ

خیر: زیرا نمیتوان گفت که با دوبرابر کردن تابع  $h_1$  تابع دیگر admissible می ماند. مثال نقض زیر را توجه کنید.



همانطور که میبینید دیگر به یک پاسخ optimal دست نیافته ایم.

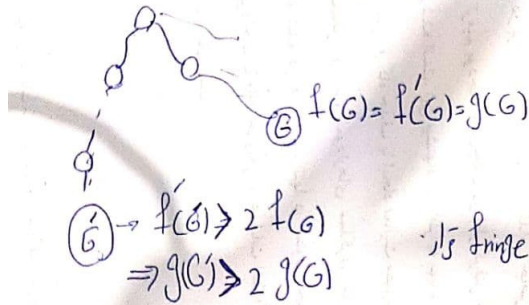
ب

درست است  
دقت کنید که تابع  $H1$  مونوتونیک است و  $H2 = 2H1$  مونوتونیک نیست زیرا زمانی به یک goal بهینه نمیرسیم که تابع  $H$  مونوتونیک نباشد. (چون رسم درخت در لثک ممکن نیست روی برگه نوشته و قرار می‌دهم.)

$$\begin{aligned} f(n) &= g(n) + h(n) \quad \text{تابع} \\ f'(n) &= g(n) + h_1(n) + h_2(n) = g(n) + 2h_1(n) \end{aligned}$$

بهان خلف:

استرا در فکلی بگوید که  $h_1$  مونوتونیک و  $h_2$  مونوتونیک نی باشد (اگر  $h_2 = 2h_1$  مونوتونیک باشد).  
ب goal آتیالی/سم.



فرض کنید که  $f(G) > 2f'(G)$  باشد.

حال نودی که به  $G$  منتمی شود در fringe قرار دارد را در فکلی بگوید. به نود  $\alpha$

$$f(\alpha) = g(\alpha) + h_1(\alpha) < g(G)$$

$$f'(\alpha) = g(\alpha) + 2h_1(\alpha) < 2g(G)$$

$$\Rightarrow f'(\alpha) < g(G')$$

بنابراین  $\alpha$  expand شود نه  $G'$   $\Leftarrow$  حال نباید استرا به یک goal خواهیم

رکیز که  $g(g)$  حداقلی در پای جواب بهینه باشد  $\Leftarrow$  زیرا در این صورت آن goal expand نمی‌شود

ج

خیر: مانند قسمت الف (به مثال نقض قسمت الف توجه کنید).

## سوالات جست و جوی محلی

۱

زمانی که از نظر حافظه محدودیت داشته باشیم همچنین رسیدن به یک global optimal goal لازم نباشد و یک sub optimal goal برایمان کافی باشد. (البته با روش های مختلف مطرح شده در کلاس می توان این sub optimal goal را به global optimal goal نزدیک و نزدیک تر کرد.)  
الگوریتم های search local معمولاً بسیار سریع هستند در مقایسه با  $A^*$  از طرفی اما local search ها ممکن است در یک sub optimal goal گیر کنند و یا حتی ممکن است که در یک جواب نامطلوب مانند مساله ۸ وزیر کانورج کنند. (البته متد های مختلفی برای برون رفت از این حالت وجود دارد) اما در حالت کلی ممکن است الگوریتم های جست و جوی محلی به یک جواب sub optimal کانورج کنند نه یک global optimal همچنین ممکن است که اختلاف جواب بدست آمده با جواب کاملاً بهینه زیاد باشد.

۲

آ

الگوریتم مانند hill-climbing search عمل خواهد کرد.

ب

نمی توان اسم الگوریتمی را روی آن گذاشت. مانند یک BFS است که در همان مرحله ابتدا تمام نود های ممکن را تولید می کند (ترتیب دیگر موضوعیت ندارد یک درخت که حداقل به تعداد تمام نود های BFS همان ابتدا استیت ها را تولید می کند). از آنجا که یکی از استیت ها جواب بهینه است همان انتخاب می شود.

ج

اگر قسمت شرط terminate را در نظر نگیریم (زیرا در غیر آن صورت همان استیت اول ریتن می شود.) الگوریتم به greedy hill-climbing تبدیل می شود. زیرا در هر مرحله اجازه حرکت نابهینه را نمی دهد.

د

کاهش سریع دما به آن معناست که  $T$  را به سرعت (زود) صفر نزدیک کنیم در آن صورت الگوریتم مذکور به hill climbing نزدیک و نزدیک تر می شود می شود.  
اگر دما را مقدار مثبت و ثابت در نظر بگیریم  $T = 0$  هیچ وقت اتفاق نمی افتد و الگوریتم پایان نمی پذیرد. (در لوپ می افتد) اما اگر فرض کنیم که شرط terminate را تغییر داده و یک ماکسیم تعداد حرکت توصیف کنیم هر بار اگر در یک نقطه optimal sub گیر کرده باشیم و هنوز بتوانیم ادامه دهیم الگوریتم با یک احتمال ثابت استیت رندم را انتخاب می کند. اگر  $T$  یک عدد بزرگ باشد آنگاه الگوریتم به random walk نزدیک می شود، اما اگر یک عدد کوچک باشد به hill climbing نزدیک می شود.