بسمه تعالی

هوش مصنوعی

حل مسئله – ۶

نیمسال اول ۱۴۰۴-۱۴۰۳

دکتر مازیار پالهنگ

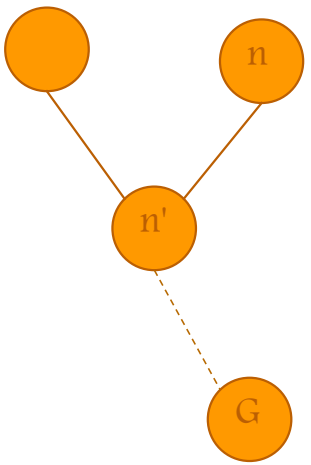
آزمایشگاه هوش مصنوعی

دانشکدهٔ مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

- جستجوی آگاهانه
- جستجوی بهترین نخست حریصانه
- حستجوی A*

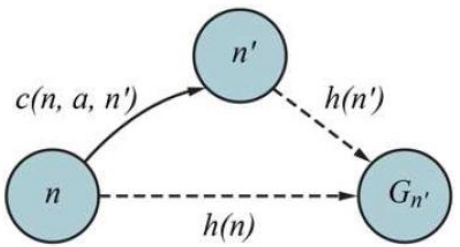


- چه اتفاقی می افتد اگر $f(n') < f(n)$ ؟
- فرض h قابل پذیرش باشد، n' قبلاً دیده شده و مسیر بهینه از طریق n و n' باشد.
- جستجوی درختی مشکلی ندارد
- چون بعداً $f(n)$ کمتر شده و انتخاب خواهد شد.
- جستجوی گرافی مسیر بهینه را از دست می دهد
- چون n' به مجموعهٔ بازدید شده منتقل می شود.

- دو راه حل:
  - با توجه به مقدار f جایگزینی انجام شود.
  - همیشه مسیر بهینه ابتدا دیده شود.
- شرط دوم به شرط سازگار بودن مکاشفه می تواند برقرار شود.

مکاشفهٔ سازگار

- h سازگار است اگر برای هر رأس n، و رأس جانشین n’ از n که با انجام عمل a حاصل شده:

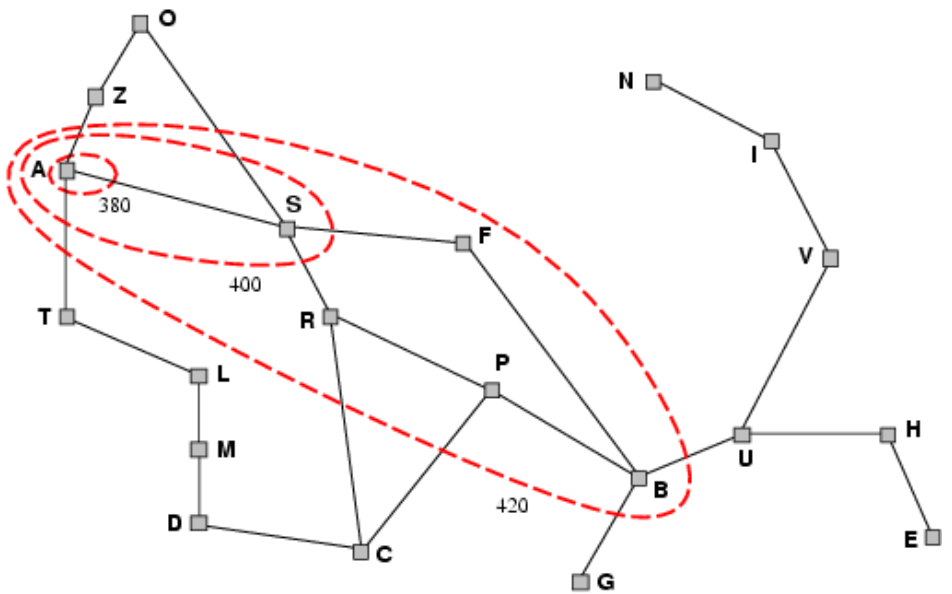


- $h(n) \leq c(n,a,n') + h(n')$
- در این صورت مقادیر f(n) در طول هر مسیری غیر کاهنده خواهند بود:
- اثبات:

$$
\begin{aligned}
g(n') &= g(n) + c(n,a,n') \\
f(n') &= g(n') + h(n') \\
&= g(n) + c(n,a,n') + h(n') \\
&\geq g(n) + h(n) \\
&\geq f(n)
\end{aligned}
$$

- بنابر این، دنبالهٔ رئوسی که توسط A^* با جستجوی گرافی بسط داده می شوند بترتیب غیرکاهندهٔ $f(n)$ می باشد.
- در واقع هر رأسی که بازدید می شود مسیر بهینه از ریشه تا این رأس یافته شده است.
- چون اگر رأسی همانند n در مسیر بهینه تا رأس بازدید شده وجود داشت حتماً زودتر مشاهده شده بود.
- از این رو اولین رأس هدف که بسط داده می شود باید بهینه باشد.
- هر مکاشفه سازگار، قابل پذیرش نیز خواهد بود.

- بدین ترتیب در فضای حالت می توانیم دوره هائی رسم نمائیم:
- دورهٔ iام دارای همهٔ رئوس با $f<=f_i$ بوده که $f_i<f_{i+1}$



- نهایتاً به fای برابر با هزینهٔ مسیر تا یک هدف خواهیم رسید لذا کامل است
- به شرط آنکه b محدود و هر مرحلهٔ هزینهٔ ای بیش از ε دارا باشد.
- A^* همهٔ رئوس با $f(n)<C^*$ را جستجو می کند.
- A^* امکان دارد تعدادی رأس با $f(n)=C^*$ را قبل از رأس هدف بازدید کند.
- مشکل حافظه هنوز در A^* وجود دارد.

- در مثال جهانگرد، A^* هیچگاه نیاز به بسط رأس ارومیه پیدا نکرد.
- چون مکاشفه قابل پذیرش است عدم جستجوی این رأس از بهینگی روش نمی کاهد.
- اصطلاحاً گفته می شود این شاخه های درخت جستجو هرس شده اند.

- A^* با مکاشفهٔ سازگار همچنین بصورت بهینه کارآ (optimally efficient) است.
- هیچ الگوریتم بهینه دیگری رئوس کمتر از آنچه A^* بسط می دهد بسط نخواهد داد (بجز احتمالاً چند رأس که $f(n)=C^*$)
- چون هر الگوریتمی که رأسی با $f(n)<C^*$ را بسط ندهد در ریسک از دست دادن حل بهینه قرار خواهد گرفت.

عمیق ساز تکراری A* (IDA*)

- A* حافظهٔ زیادی می تواند استفاده نماید.
- عمیق ساز تکراری A* همانند جستجوی عمیق ساز تکراری می باشد،
- با این تفاوت که مقدار f=g+h برای حد عمق استفاده می شود.
- در هر تکرار مقدار حد، کمترین مقدار f رأسی است که از حد عمق در مرحلهٔ قبل عبور کرده بود.
- در این حالت نیاز به استفاده از یک صف اولویت دار نمی باشد.

عمیق ساز تکراری A* (IDA*)

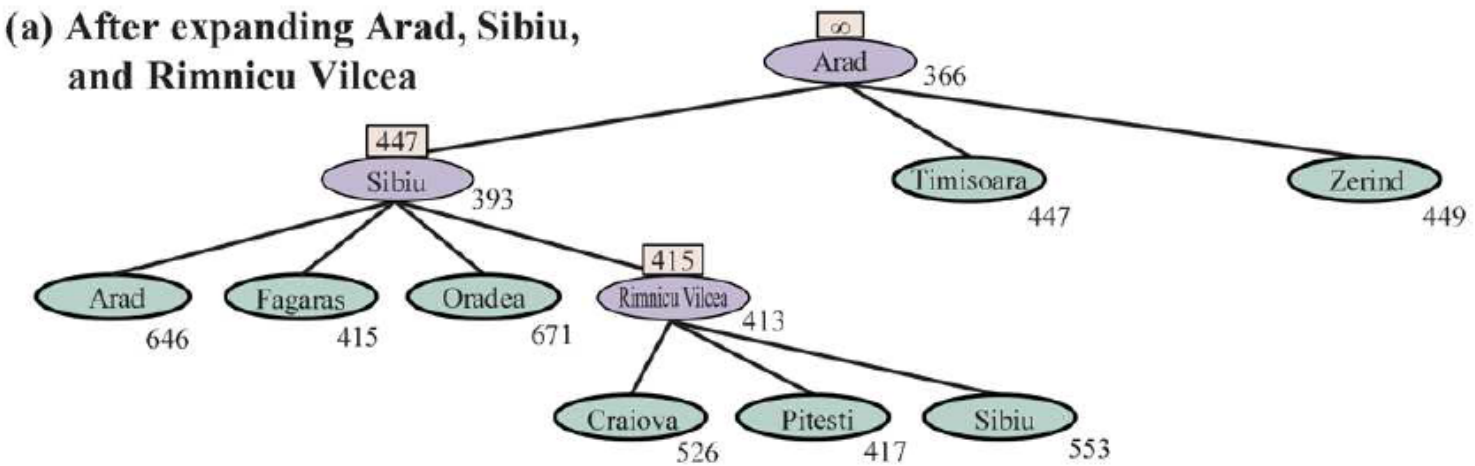
- الگوریتم همانند عمیق ساز تکراری می تواند بصورت بازگشتی پیاده سازی شود،
- فقط حد عمق با استفاده از مقدار f تعیین می شود، و
- حد بعدی هم از روی مقدار f تکرار قبل بدست می آید.
- بنابر این میزان استفادهٔ حافظهٔ این الگوریتم شبیه به روش عمق نخست می باشد.
- ولی در شرایطی می تواند میزان رئوسی که بازدید می شوند از A* بیشتر باشد.

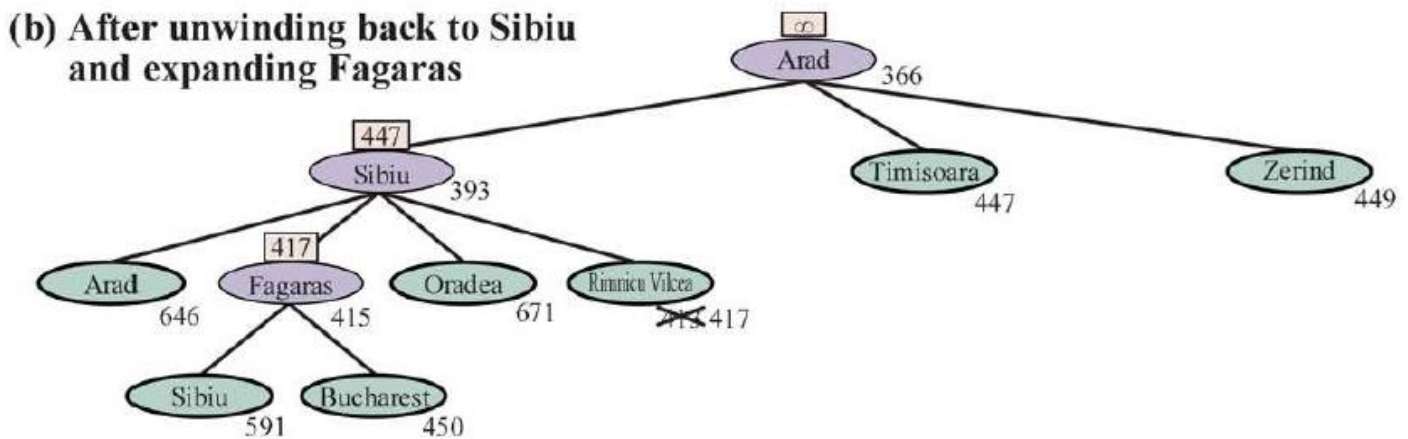
```
function IDA*(problem, max_limit) returns a solution, or failure
        root = Make-Node(problem.Initial-State)
        limit = f(root)
        while  limit < max_limit
                result, new_limit = DLS(root, limit)
                if result = solution, return solution
                limit = new_limit
        return failure

function DLS(n, limit) returns a solution, or new_limit
        if f(n) > limit return f(n)
        if problem.GoalTest(n.State) then return solution
        else
            min = ∞
            for all s in Expand(n)
                result, new_limit = DLS(s, limit)
                if result = solution return result
                else if  new_limit < min
                        min = new_limit
        return min
```

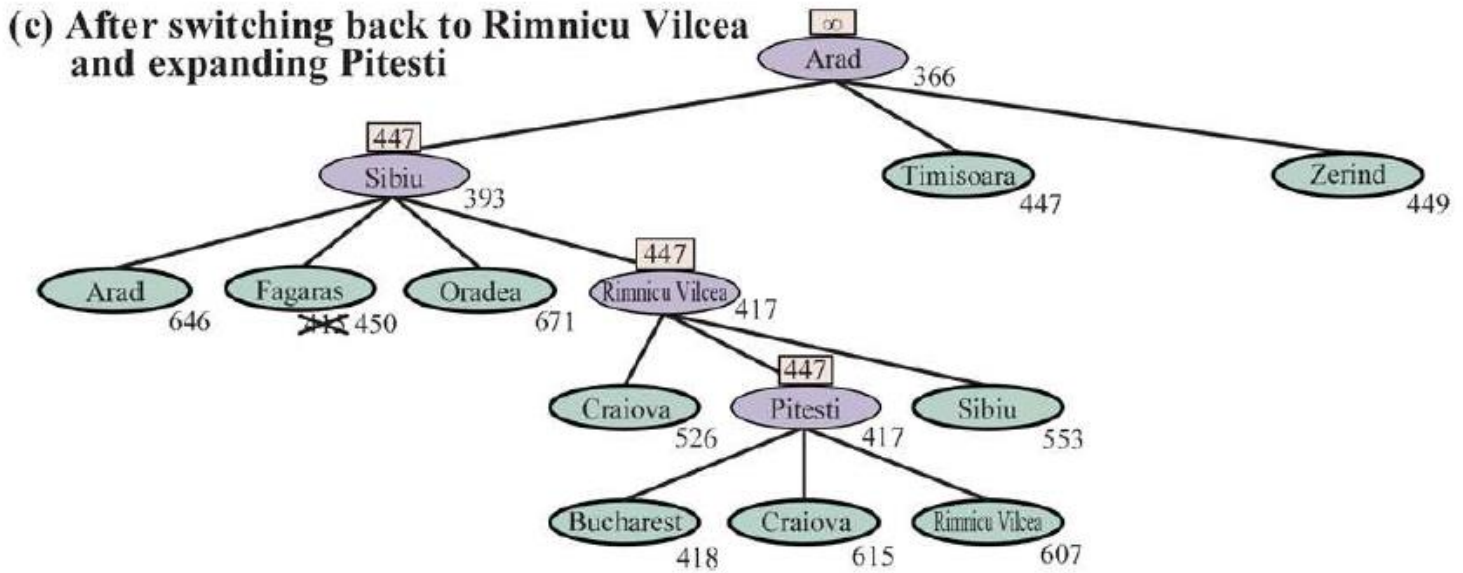
جستجوی بهترین نخست بازگشتی

- شبیه به عمق نخست بازگشتی عمق محدود شده
- الگوریتم مقدار f بهترین مسیر جایگزین از هر جد رأس فعلی را دنبال می کند.
- اگر f رأس فعلی از این حد عبور کند، بازگشت به مسیر جایگزین انجام می گیرد.

.


(a) After expanding Arad, Sibiu, and Rimnicu Vilcea
∞
Arad
366
447
Sibiu
393
Timisoara
447
Zerind
449
Arad
646
Fagaras
415
Oradea
671
415
Rimnicu Vilcea
413
Craiova
526
Pitesti
417
Sibiu
553


(b) After unwinding back to Sibiu and expanding Fagaras
∞
Arad
366
447
Sibiu
393
Timisoara
447
Zerind
449
417
Arad
646
Fagaras
415
Oradea
671
Rimnicu Vilcea
417
Sibiu
591
Bucharest
450


(c) After switching back to Rimnicu Vilcea and expanding Pitesti
∞
Arad
366
447
Sibiu
393
Timisoara
447
Zerind
449
Arad
646
Fagaras
450
Oradea
671
447
Rimnicu Vilcea
417
Craiova
526
447
Pitesti
417
Sibiu
553
Bucharest
418
Craiova
615
Rimnicu Vilcea
607

- بهینه؟
  - بله اگر h قابل پذیرش باشد
- کامل؟
  - بله
- پیچیدگی فضا $O(bd)$
- پیچیدگی زمان – مشخص نیست.

الگوریتم SMA*

- RBFS حافظهٔ زیادی استفاده نمی کند.
- SMA* همانند A* جستجو می کند تا حافظه پر شود.
- در این هنگام برگ با بزرگترین f را انداخته و مقدار f آن را در والدش ذخیره می کند.
- ممکن است رأسی در مسیر پاسخ بهینه قرار داشته باشد ولی به علت کمبود حافظه دیگر قابل بسط دادن نباشد.
- مثلاً وقتی یک رأس برگ داریم و آن به سمت هدف می رود و حافظه دیگر وجود ندارد.

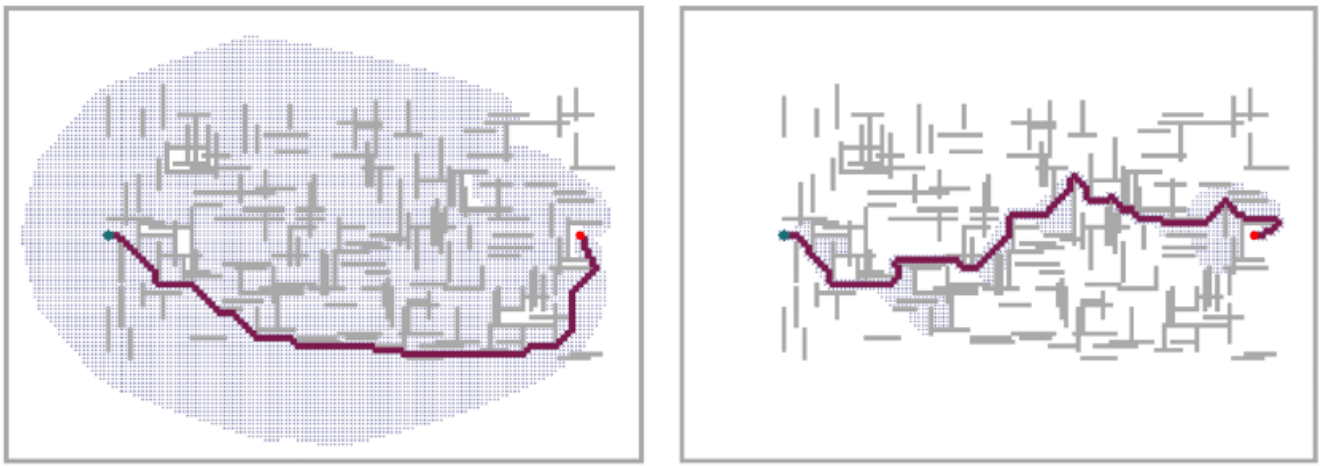
A* وزندار

- در صورتی که به پاسخ زیربهینه ولی نسبتاً خوب راضی باشیم می توان از A* وزندار استفاده کرد.
- در این صورت مکاشفه غیرقابل پذیرش است.
- در این حالت:

$$f(n) = g(n) + W \times h(n)$$

- W>1

Figure 3.21



(a) (b)

Two searches on the same grid: (a) an A* search and (b) a weighted A* search with weight $W = 2$. The gray bars are obstacles, the purple line is the path from the green start to red goal, and the small dots are states that were reached by each search. On this particular problem, weighted A* explores 7 times fewer states and finds a path that is 5% more costly.

خلاصه

- IDA*
- جستجوی بهترین نخست بازگشتی
- SMA*
- A* وزندار
- توابع مکاشفه ای


والسلام

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائهٔ شفاهی می باشد و **به هیچ وجه یک جزوهٔ درسی نیست** و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.

- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائهٔ شفاهی می باشد و **به هیچ وجه یک جزوهٔ درسی نیست** و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.