

بسمه تعالی

هوش مصنوعی

جستجو در محیطهای پیچیده - ۵

نیمسال اول ۱۴۰۳-۱۴۰۲

دکتر مازیار پالهنک

آزمایشگاه هوش مصنوعی

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

یادآوری

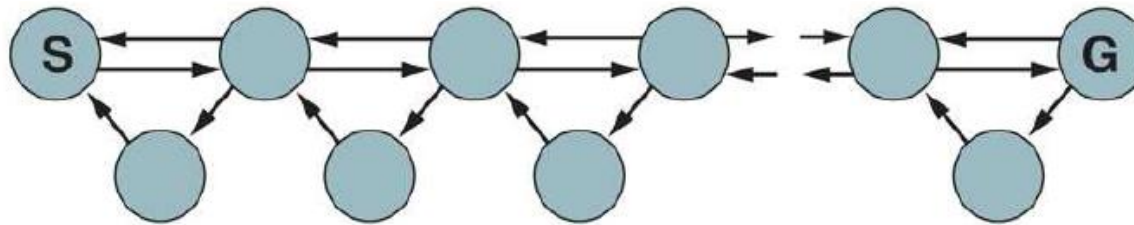
- الگوریتمهای جستجوی محلی
 - حالت فعلی را نگهدار - سعی کن آن را بهبود دهی
- جستجوی تپه نوردی
- سردشدن شبیه سازی شده
- جستجوی پرتو محلی، و تصادفی، الگوریتم ژنتیک
- جستجوی محلی در فضای پیوسته
- جستجو با اعمال قطعی (پاسخ یک دنباله)
- جستجو با اعمال غیر قطعی
 - حل یک طرح شرطی، استفاده از درخت AND-OR با استفاده از فضای حالت
- جستجو برای عامل بدون حسگر
 - جستجو در فضای باور همانند حالت مشاهده پذیر با تعمیم تعریف اجزاء مسئله
- جستجو در محیط نیمه مشاهده پذیر
 - حل یک طرح شرطی، استفاده از درخت AND-OR با استفاده از فضای باور
- تجستجوی برخط
- مسائل جستجوی برخط با فرض محیط مشاهده پذیر قطعی
 - عدم توانایی مشخص کردن $\text{Result}(s,a)$ از قبل
 - نسبت رقابتی
- عامل جستجوی برخط عمق نخست Online_DFS_Agent

جستجوی محلی بر خط

- جستجوی تپه نوردی نیز همانند عمق نخست حالت محلی را دارد.
- مشکل بهینه محلی برای اکتشاف
- نمی توان باز شروع تصادفی داشت.
- گام زدن تصادفی
- انتخاب تصادفی یکی از اعمال
- می توان ثابت کرد که گام زدن تصادفی نهایتاً هدف را یافته (یا اکتشاف را کامل می کند).

■ گاهی خیلی طولانی

Figure 4.22



An environment in which a random walk will take exponentially many steps to find the goal.

■ می توان از بسط تپه نوردی با حافظه استفاده کرد.

■ ذخیره بهترین تخمین فعلی تا هدف $H(s)$

■ شروع با $h(s)$

■ اصلاح آن حین اکتشاف

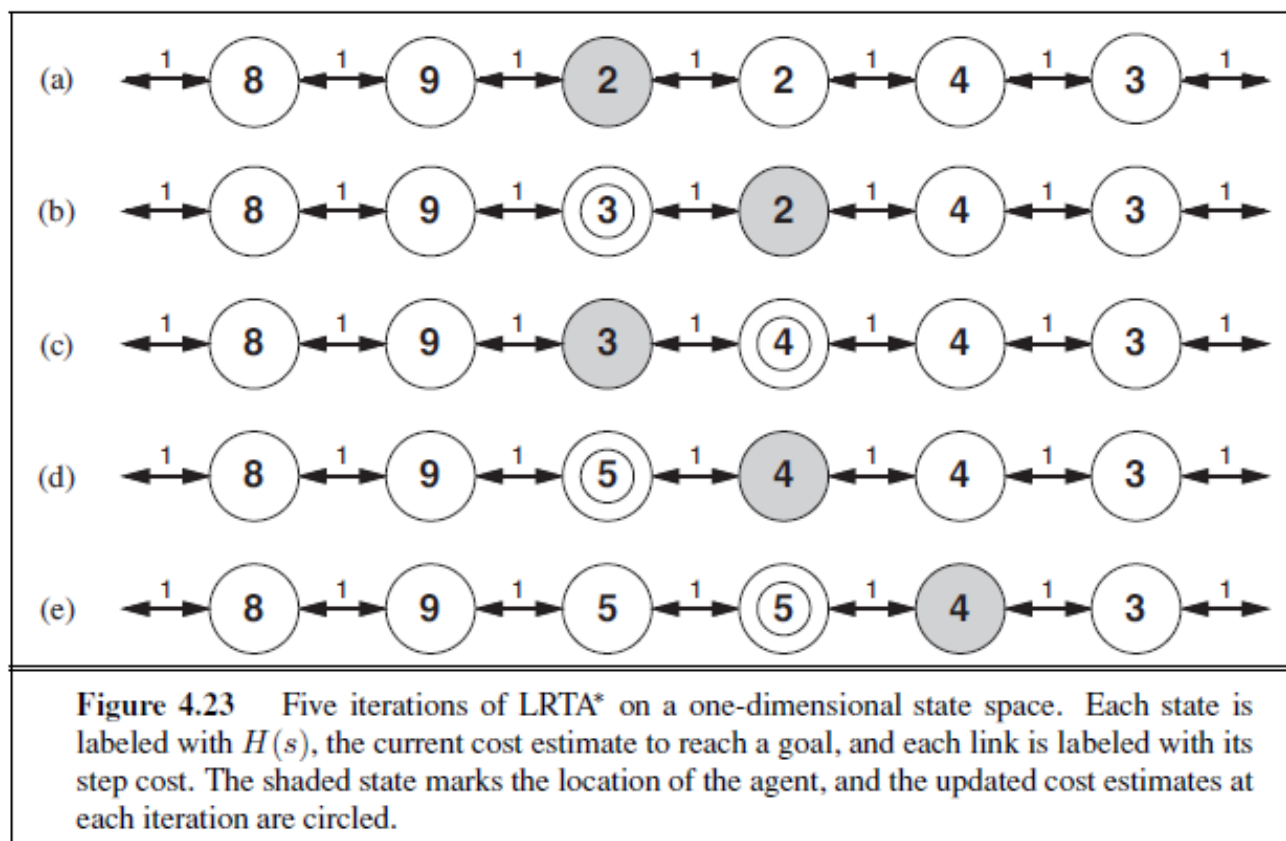


Figure 4.23 Five iterations of LRTA* on a one-dimensional state space. Each state is labeled with $H(s)$, the current cost estimate to reach a goal, and each link is labeled with its step cost. The shaded state marks the location of the agent, and the updated cost estimates at each iteration are circled.

Figure 4.24

function LRTA*-AGENT(*problem*, s' , *h*) **returns** an action
 s, *a*, the previous state and action, initially null
 persistent: *result*, a table mapping (*s*, *a*) to s' , initially empty
 H, a table mapping *s* to a cost estimate, initially empty

if IS-GOAL(s') **then return** *stop*
 if s' is a new state (not in *H*) **then** $H[s'] \leftarrow h(s')$
 if *s* is not null **then**
 $result[s, a] \leftarrow s'$
 $H[s] \leftarrow \min_{b \in \text{ACTIONS}(s)} \text{LRTA}^*\text{-COST}(s, b, result[s, b], H)$
 $a \leftarrow \operatorname{argmin}_{b \in \text{ACTIONS}(s)} \text{LRTA}^*\text{-COST}(problem, s', b, result[s', b], H)$
 $s \leftarrow s'$
 return *a*

function LRTA*-COST(*problem*, *s*, *a*, s' , *H*) **returns** a cost estimate
 if s' is undefined **then return** $h(s)$
 else return $problem.\text{ACTION-COST}(s, a, s') + H[s']$

LRTA*-AGENT selects an action according to the values of neighboring states, which are updated as the agent moves about the state space.

- الگوریتم $LRTA^*$ ضمانت می کند که در محیطهای محدود و قابل اکتشاف امن هدف را بیابد.
- برای فضاهای حالت نامحدود کامل نیست.



- دقت نمائید که پاورپوینت ابزاری جهت کمک به یک ارائه شفاهی می باشد و به هیچ وجه یک جزوه درسی نیست و شما را از خواندن مراجع درس بی نیاز نمی کند.
- لذا حتماً مراجع اصلی درس را مطالعه نمائید.
- حضور فعال در کلاس دارای امتیاز است.