



هوش مصنوعی

حسین کارشناس

دانشکده ریاضی

ترم اول ۹۴ - ۹۳

جستجو در مسائل پیچیده‌تر

جستجو در مسائل پیچیده‌تر

- حذف فرضیات ساده‌کننده مسأله
 - محیط‌های غیرقطعی
 - محیط‌های مشاهده‌پذیر جزئی
 - محیط‌های ناشناخته
- جستجو با کنش‌های غیرقطعی
- جستجو با مشاهدات جزئی
- جستجوی برخط (online) در محیط‌های ناشناخته

جستجو با کنش‌های غیر قطعی

جستجو با کنش‌های غیرقطعی

- در محیط‌های قطعی عامل بدون ادراک محیط، نتیجه کنش خود را می‌داند
- در محیط‌های غیرقطعی ادراک اهمیت پیدا می‌کند
 - کنش‌های آینده به ادراک‌های آینده بستگی خواهند داشت
 - راه‌حل بجای یک دنباله از کنش‌ها، یک برنامه احتیاطی (contingency plan) یا یک راهبرد (strategy) برای حالت‌های ممکن مختلف است
- دنیای جاروبرقی نامنظم (erratic)
 - اعمال مکش روی یک خانه کثیف ممکن است خانه کناری را نیز تمیز کند
 - اعمال مکش روی یک خانه تمیز ممکن است آن را کثیف کند

جستجو با کنش‌های غیر قطعی

- نتیجه اعمال یک کنش در یک حالت مجموعه‌ای از حالات ممکن

است: $RESULTS(s, a) = \{s_1, \dots, s_n\}$

- راه‌حل‌ها درخت هستند نه دنباله

- بستگی کنش‌ها به حالت ادراک شده: دنباله شرطی از کنش‌ها $\leftarrow$ درخت

- درخت‌های AND-OR

- گره‌های OR: گره‌های معمولی در درخت‌های جستجو شامل حالت‌های محیط

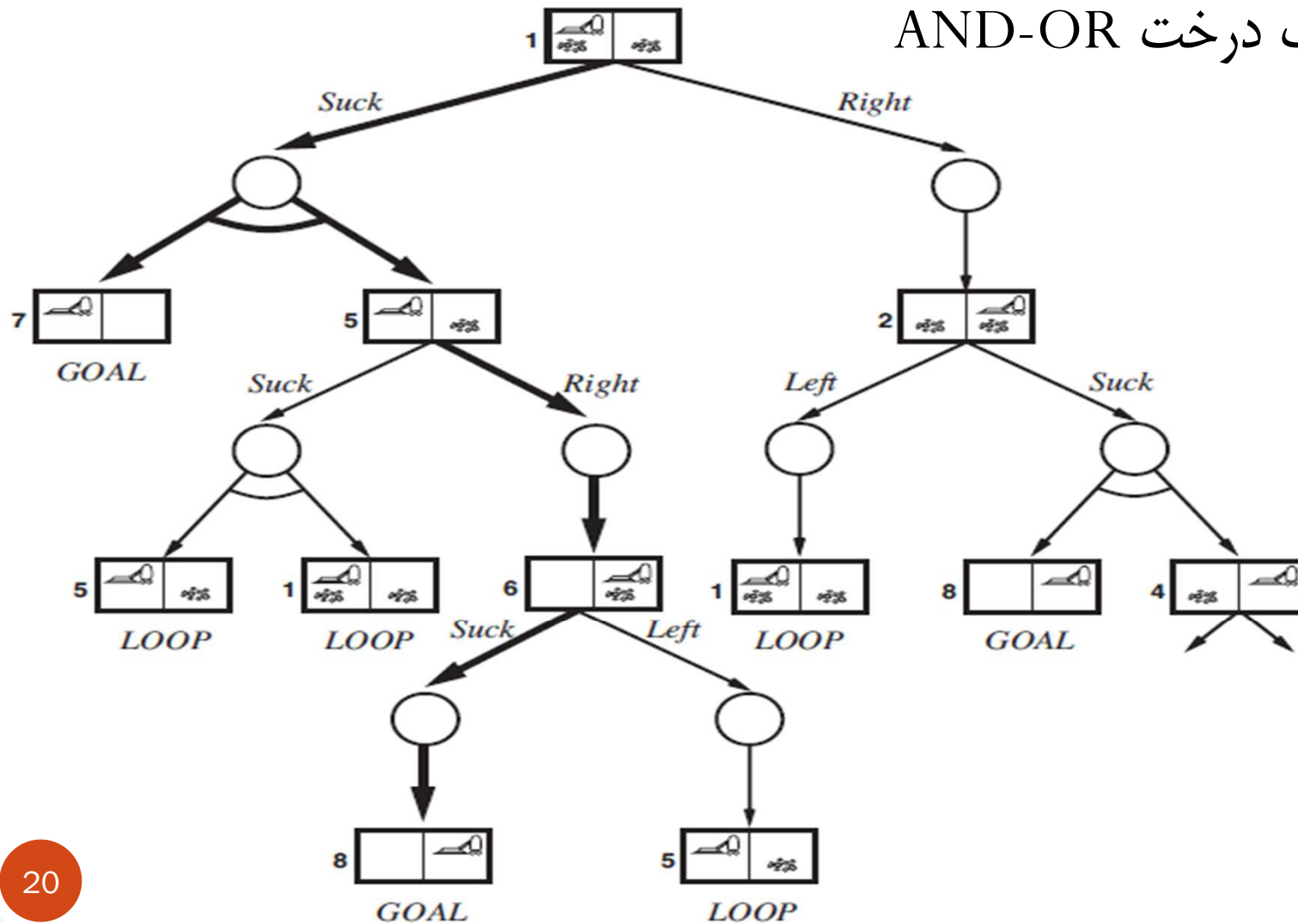
• براساس کنش‌های ممکن عامل در یک حالت منشعب می‌شوند

- گره‌های AND: در بین سطوح حاوی گره‌های OR قرار می‌گیرند و تمام

خروجی‌های ممکن از اعمال یک کنش در محیط را نشان می‌دهند

جستجو با کنش‌های غیر قطعی

● مثال: یک درخت AND-OR



جستجو با کنش‌های غیر قطعی

- یک الگوریتم جستجو بر مبنای جستجوی عمق اول

function AND-OR-GRAPH-SEARCH(*problem*) **returns** a conditional plan, or failure
OR-SEARCH(*problem*.INITIAL-STATE, *problem*, [])

function OR-SEARCH(*state*, *problem*, *path*) **returns** a conditional plan, or failure
if *problem*.GOAL-TEST(*state*) **then return** the empty plan
if *state* is on *path* **then return** failure
for each *action* **in** *problem*.ACTIONS(*state*) **do**
 plan $\leftarrow$ AND-SEARCH(RESULTS(*state*, *action*), *problem*, [*state* | *path*])
 if *plan* $\neq$ failure **then return** [*action* | *plan*]
return failure

function AND-SEARCH(*states*, *problem*, *path*) **returns** a conditional plan, or failure
for each s_i **in** *states* **do**
 *plan*_{*i*} $\leftarrow$ OR-SEARCH(s_i , *problem*, *path*)
 if *plan*_{*i*} = failure **then return** failure
return [if s_1 then *plan*₁ else if s_2 then *plan*₂ else ... if s_{n-1} then *plan*_{*n-1*} else *plan*_{*n*}]

جستجو با کنش‌های غیر قطعی

- زیردرختی یک راه‌حل برای یک مسأله AND-OR است که:

1. هر برگ آن یک گره هدف باشد (یا حاوی گره هدف باشد)

2. در هر یک از گره‌های OR یک کنش را مشخص کند

3. در هر یک از گره‌های AND، تمام شاخه‌ها را در نظر بگیرد

- دنیای جاروبرقی لغزنده (slippery)

- بعضی مواقع کنش‌های حرکت به چپ و راست به درستی اعمال نمی‌شوند

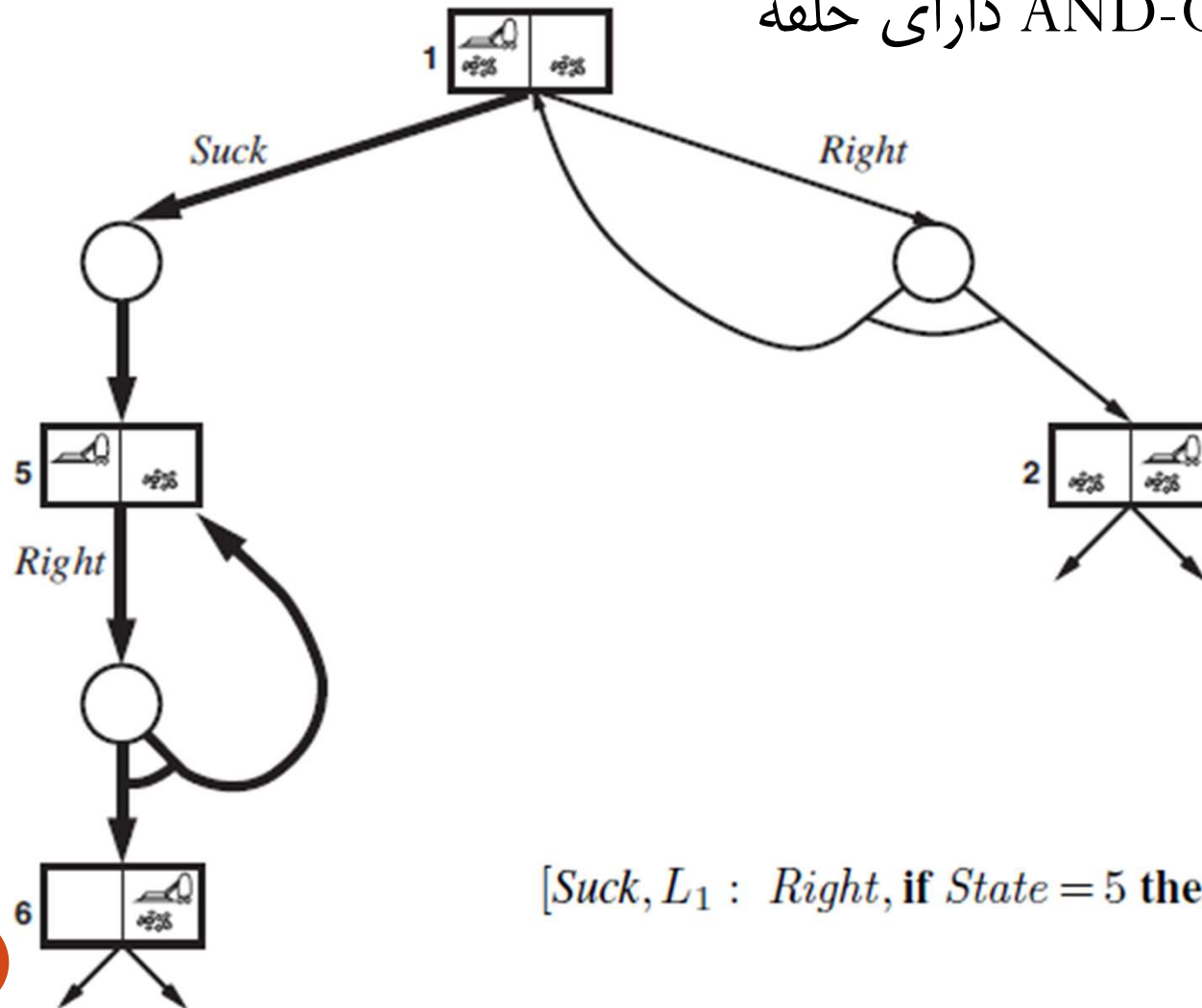
- راه‌حل غیر حلقه‌ای ندارد

- راه‌حل‌های حلقه‌ای: در بردارنده‌ی چند حلقه از کنش‌ها و حالت‌ها

- استفاده از برچسب‌ها برای ساده‌سازی پیاده‌سازی

جستجو با کنش‌های غیر قطعی

• مثال: درخت AND-OR دارای حلقه



• راه حل:

$[Suck, L_1 : Right, \text{if } State = 5 \text{ then } L_1 \text{ else } Suck]$

جستجو با مشاهدات جزئی

جستجو با مشاهدات جزئی

- ادراک عامل برای تعیین دقیق حالت فعلی کافی نیست
- حالت باور (belief state): مجموعه‌ای از حالت‌های ممکن که عامل، براساس دنباله‌ی کنش‌ها و دریافت‌هایش تا آن لحظه، باور دارد که در آن قرار دارد
- جستجو بدون مشاهده: مسأله بدون حسگر (sensorless) یا پیرو (conformant)
- ادراکی از محیط وجود ندارد
- آیا حل مسأله در این حالت امکان‌پذیر است؟ بله
- اجبار (coercion): بردن محیط به یک حالت خاص

جستجو بدون مشاهده

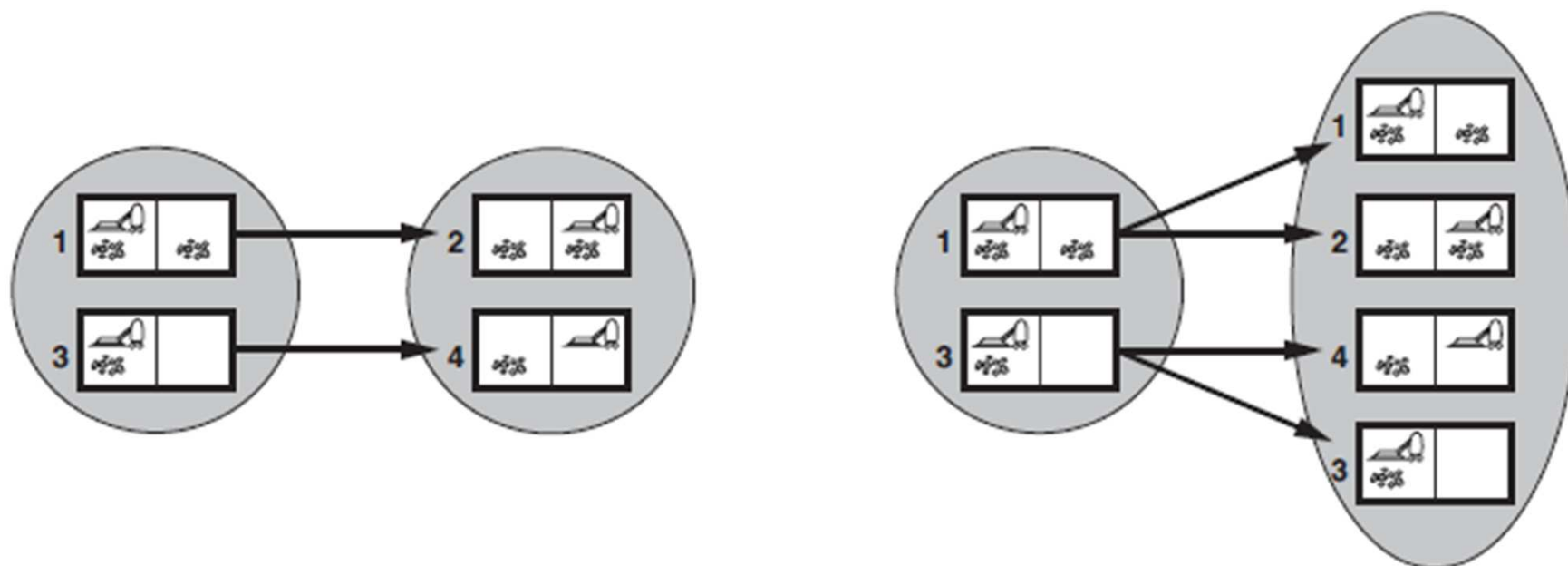
- دنیای جاروبرقی بدون حسگر
- حالت (باور) اولیه: مجموعه تمام حالت‌های ممکن مسأله
- حل مسأله با جستجو در فضای حالت باور
 - در این فضا مسأله کاملاً مشاهده‌پذیر است
 - عامل همیشه از باور خود اطلاع دارد
 - راه‌حل دنباله‌ای از کنش‌هاست
 - هیچ ادراکی از محیط وجود ندارد ← هیچ برنامه احتیاطی لازم نیست
- تعریف یک مسأله بدون مشاهده از روی مسأله واقعی
 - حالت‌های باور: تمام زیرمجموعه‌های ممکن از حالت‌های مسأله واقعی
 - حالت باور اولیه: معمولاً مجموعه تمام حالت‌های ممکن مسأله

جستجو بدون مشاهده

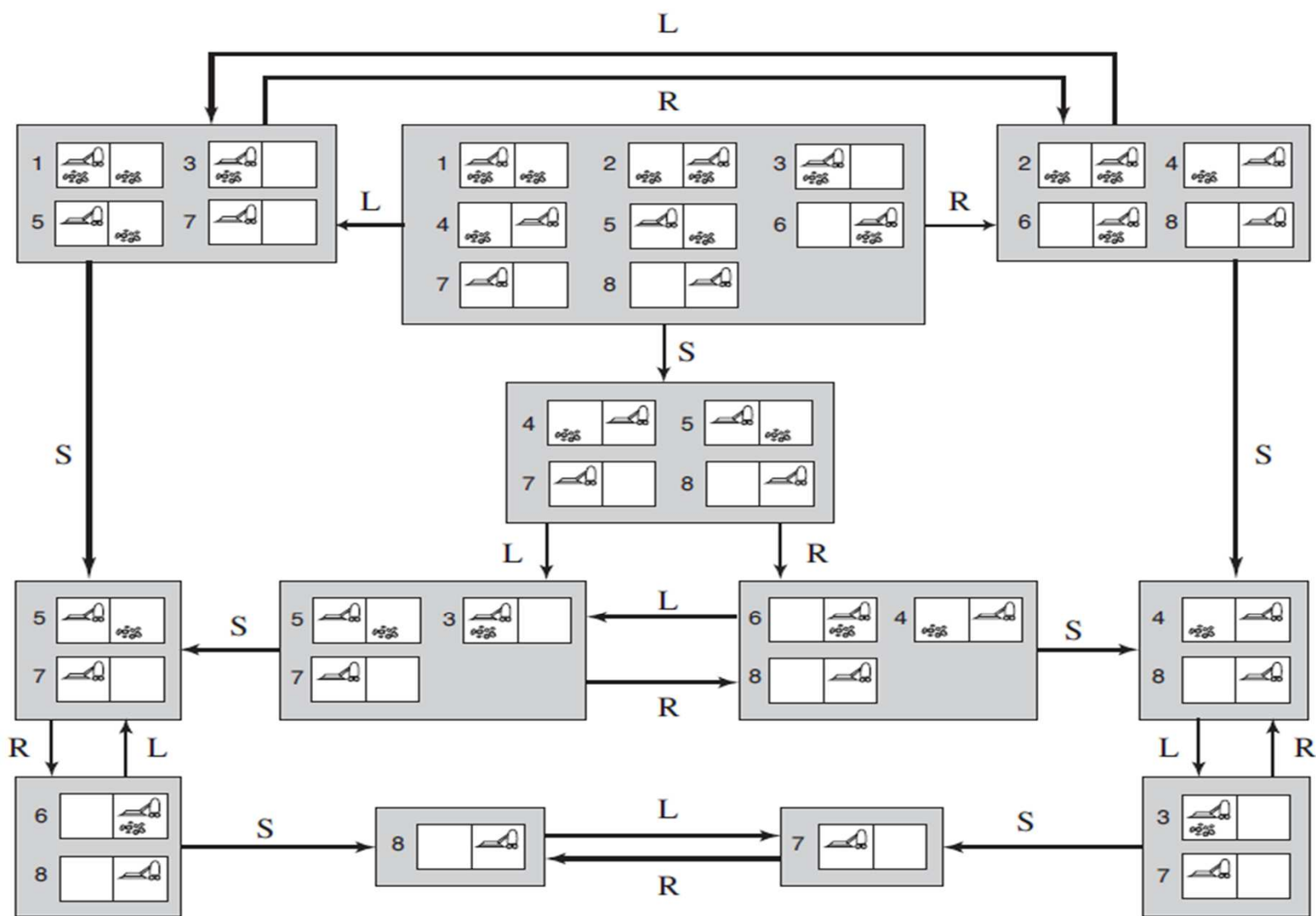
- تعریف یک مسأله بدون مشاهده از روی مسأله واقعی (ادامه)
- کنش‌ها: اجتماع یا اشتراک تمام کنش‌های ممکن در هر یک از حالات موجود در یک حالت باور
- مدل انتقال: مجموعه تمام حالت‌های قابل دستیابی از اعمال یک کنش به هر یک از حالات موجود در یک حالت باور
 - اگر کنش‌ها غیر قطعی باشند حالت باور جدید می‌تواند بزرگتر از حالت باور قبلی باشد
 - پیش‌بینی: تولید یک حالت باور جدید با اعمال یک کنش بر یک حالت باور دیگر
- آزمایش هدف: رسیدن به حالت باوری که تمام حالات واقعی آن هدف باشند
 - ممکن است عامل بدون اینکه بداند قبلاً به یک حالت هدف واقعی رسیده باشد
- هزینه مسیر: فرض می‌شود هزینه اعمال یک کنش در تمام حالت‌های یک حالت باور یکسان باشد

جستجو بدون مشاهده

- مثالی از مدل انتقال برای دنیای جاروبرقی بدون مشاهده با کنش‌های قطعی و غیرقطعی



جستجو بدون مشاهده



جستجو بدون مشاهده

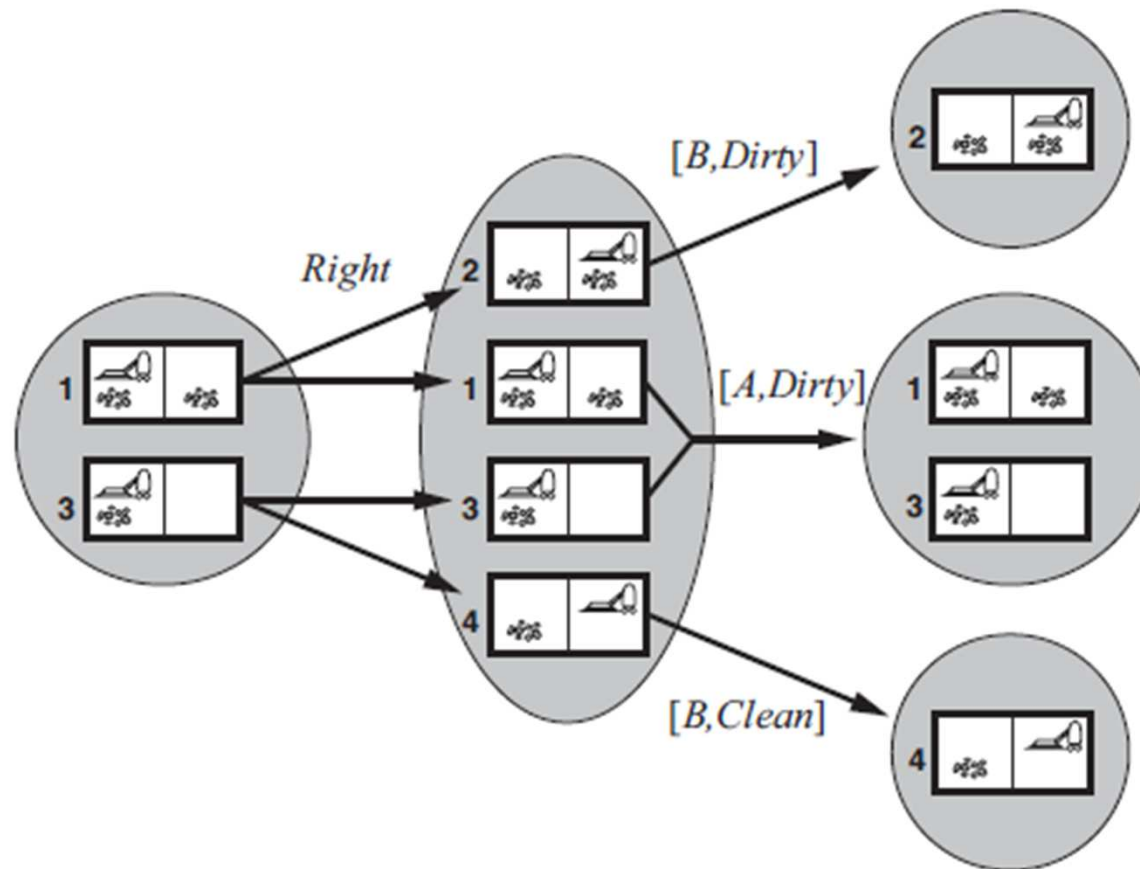
- امکان استفاده از تمام الگوریتم‌های جستجو برای جستجو در فضای حالت‌های باور
- اگر با یک دنباله از کنش‌ها از یک حالت باور به هدف برسیم، می‌توان با همان دنباله از تمام زیرمجموعه‌های آن حالت باور هم به حالت هدف رسید
- امکان هرس کردن بعضی از شاخه‌های مرتبط با زیرمجموعه‌های حالت باور
- دو مشکل اصلی در جستجوی فضای باور
 - تعداد زیاد حالت‌های باور ممکن: 2^n
 - تعداد زیاد حالت‌های واقعی در هر حالت باور
 - استفاده از نمایش‌های کارآمدتر
- جستجوی حالت باور افزایشی ← امکان شناسایی سریع شکست

جستجو با مشاهدات جزئی

- تفاوت اصلی با وضعیت بدون مشاهده در مدل انتقالی است
- در حالت کلی انتقال از یک حالت باور به حالت دیگر در سه مرحله صورت می‌گیرد
 1. پیش‌بینی: تولید یک حالت باور با اعمال یک کنش بر حالت باور فعلی
 2. پیش‌بینی مشاهده: ادراک‌های قابل مشاهده در حالت باور جدید در نظر گرفته می‌شوند (اگر ادراک مشخص نباشد)
 3. به‌نگام‌سازی: حالت‌های باور جدیدی که با مشاهده هر یک از ادراک‌های پیش‌بینی شده از حالت باور تولید شده قابل دستیابی هستند
- مشاهده باید تردید در مورد حالت‌های موجود در حالت باور را کم کند

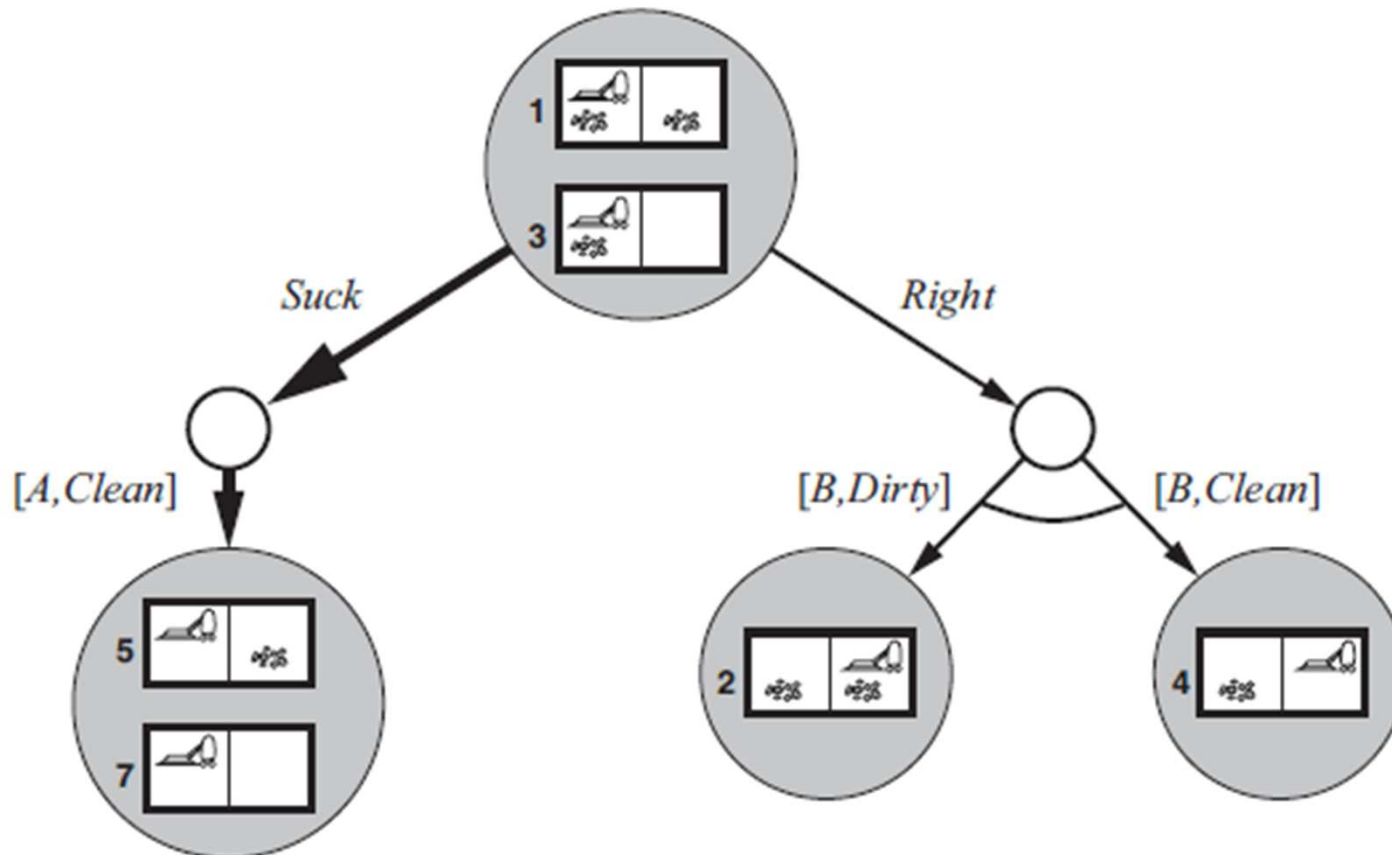
جستجو با مشاهدات جزئی

- مدل انتقال در دنیای جاروبرقی لغزنده



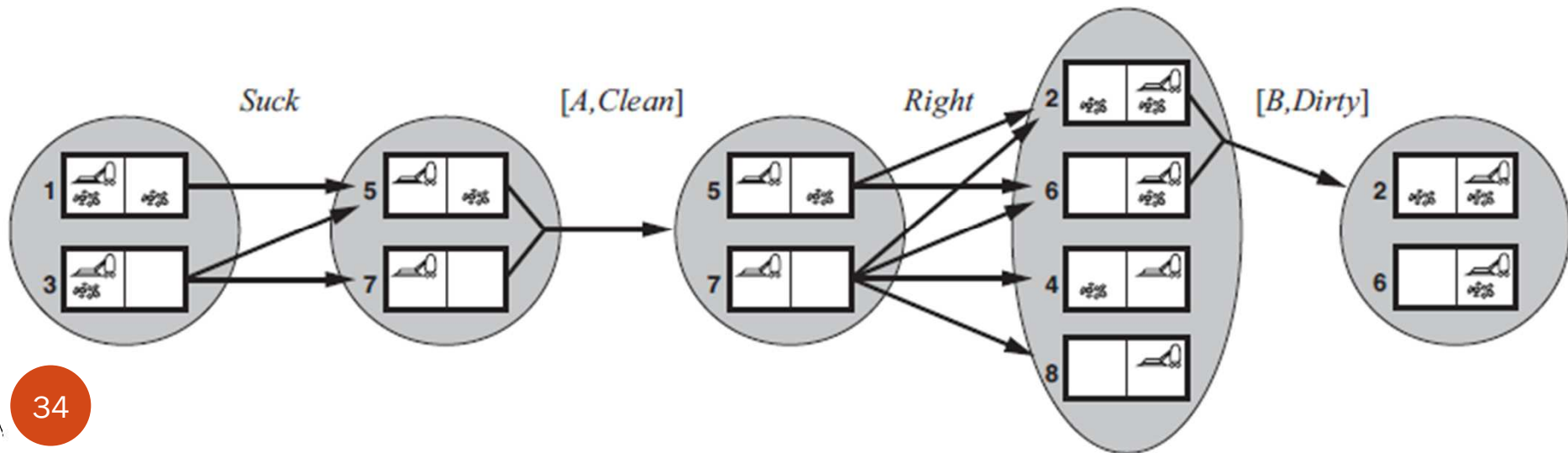
جستجو با مشاهدات جزئی

- حل مسائل مشاهده‌پذیر جزئی با جستجوی AND-OR
- گره‌های OR شامل حالت‌های باور بجای حالت‌های واقعی



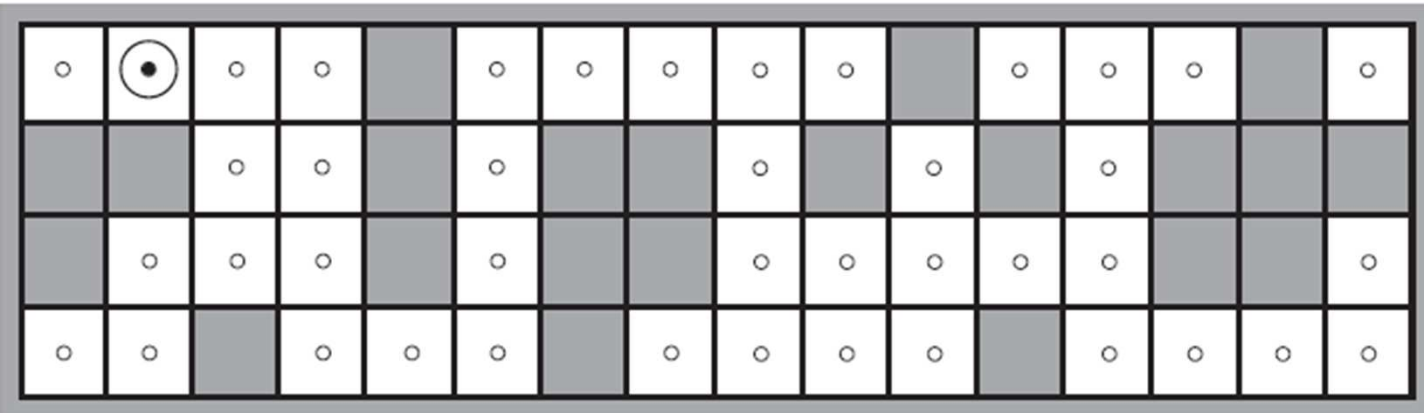
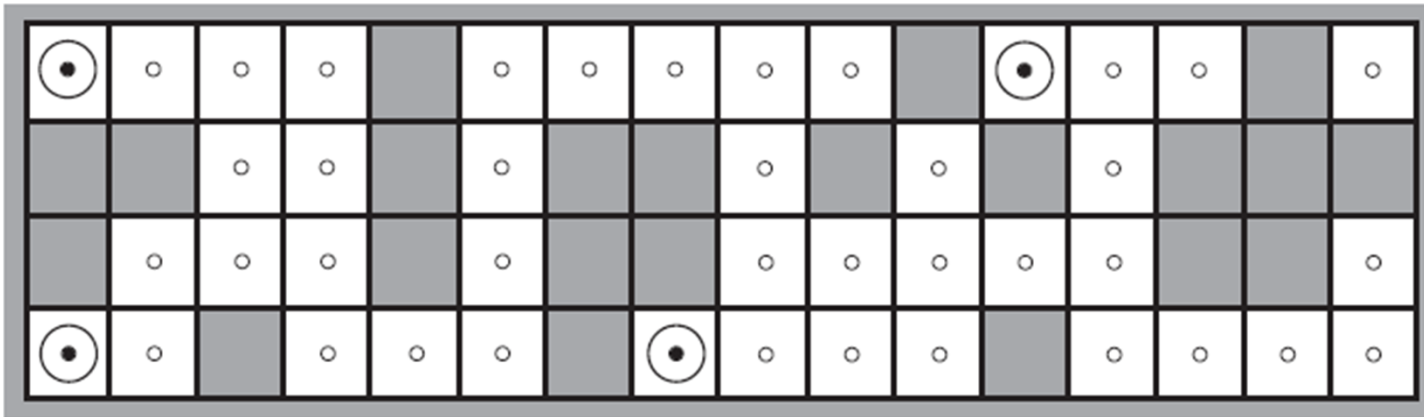
جستجو با مشاهدات جزئی

- حلقه پیش‌بینی – به‌نگام‌سازی در حل این مسأله
- $b' = \text{Update}(\text{Predict}(b, a), o)$
- دنیای جاروبرقی مهد کودک (kindergarten)
- هر خانه‌ای ممکن است در هر زمانی بجز هنگامی که در حال تمیز شدن است کثیف شود



جستجو با مشاهدات جزئی

- مثال: موقع یابی (localization) ربات با دو دریافت: NS و NSW



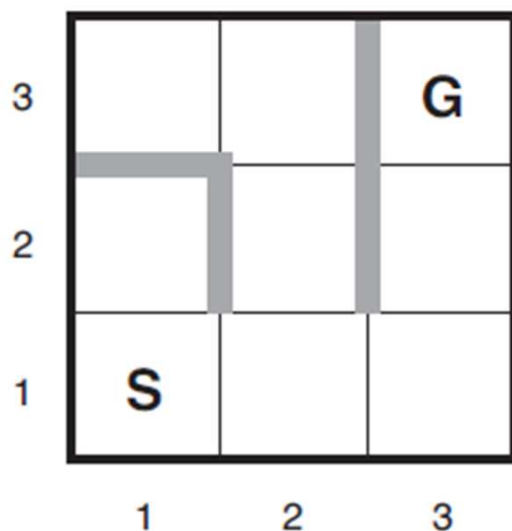
جستجوی بر خط در محیط‌های ناشناخته

جستجوی برخط در محیط‌های ناشناخته

- جستجوی برخط یا برون از خط (offline)
- جاگذاری (interleaving) جستجو و اجرا
- در محیط‌های پویا و نیمه‌پویا
- در محیط‌های غیرقطعی ← تمرکز بر حالات باور ممکن بجای تمام حالات باور
- در محیط‌های ناشناخته ضروری است
- مسأله کاوش (exploration)
- مثال
- تهیه نقشه از یک ساختمان جدید توسط یک ربات برای مسیریابی بین دو نقطه
- نوزاد تازه متولد شده

مسائل جستجوی بر خط

- با فرض قطعی بودن و کاملاً مشاهده‌پذیر بودن محیط
- نحوه تعریف مسأله



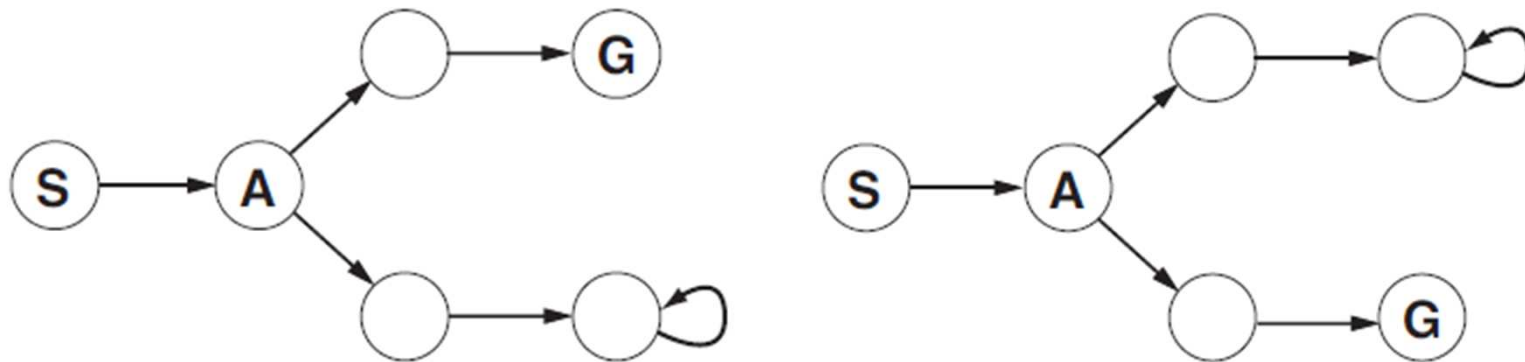
- کنش‌های ممکن در هر حالت
- آزمایش هدف روی هر یک از حالت‌ها
- هزینه‌ی هر یک از گام‌ها
- حالت‌های محیط ناشناخته هستند
- مدل انتقال ناشناخته است
- نتیجه اعمال یک کنش در یک حالت تا انجام ندادن آن کنش در آن حالت مشخص نمی‌شود

مسائل جستجوی بر خط

- امکان دسترسی به یک تابع اکتشافی پذیرفتنی وجود دارد
- هدف رسیدن به یک حالت هدف با کمترین هزینه است
- هدف می‌تواند کاوش کل محیط باشد
- هزینه مسیری که عامل طی می‌کند، هزینه‌ی آن را مشخص می‌کند
- نسبت رقابتی (competitive ratio): هزینه مسیر بر خط نسبت به کوتاه‌ترین هزینه مسیر
- می‌تواند نامتناهی باشد
- بن‌بست‌ها
- هیچ الگوریتمی جستجوی برخطی نمی‌تواند در تمام فضاها حالت از بن‌بست جلوگیری کند

مسائل جستجوی برخط

- مثالی از دو فضای حالتی که جستجوی برخط را به بن بست می کشاند



- فرض می شود فضای حالت به صورت مطمئن کاوش پذیر (safely explorable) باشد

- از تمام حالت ها امکان دسترسی به حالت هدف وجود دارد
- نسبت رقابتی می تواند همچنان نامحدود باشد

جستجوی بر خط

- تفاوت با جستجوی برون از خط
- بسط واقعی گره‌ها بجای شبیه‌سازی عمل بسط
- امکان پرش در فضای حالت برای بسط گره‌ها وجود ندارد ← نیاز به بسط به ترتیب موضعی
- جستجوی عمق اول
- جستجوی موضعی
- ساختن مدل انتقال یا نقشه محیط به صورت افزایشی در حین جستجو

جستجوی بر خط

• جستجوی عمق اول بر خط

function ONLINE-DFS-AGENT(s') **returns** an action
 inputs: s' , a percept that identifies the current state
 persistent: *result*, a table indexed by state and action, initially empty
 untried, a table that lists, for each state, the actions not yet tried
 unbacktracked, a table that lists, for each state, the backtracks not yet tried
 s, *a*, the previous state and action, initially null

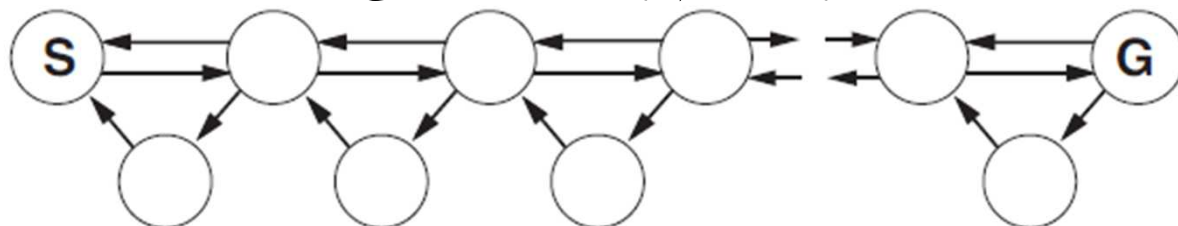
if GOAL-TEST(s') **then return** *stop*
 if s' is a new state (not in *untried*) **then** *untried*[s'] $\leftarrow$ ACTIONS(s')
 if *s* is not null **then**
 result[*s*, *a*] $\leftarrow s'$
 add *s* to the front of *unbacktracked*[s']
 if *untried*[s'] is empty **then**
 if *unbacktracked*[s'] is empty **then return** *stop*
 else $a \leftarrow$ an action b such that *result*[s' , b] = POP(*unbacktracked*[s'])
 else $a \leftarrow$ POP(*untried*[s'])
 $s \leftarrow s'$
 return *a*

جستجوی برخط

- جستجوی عمق اول برخط
 - نیاز به برگشت به اجداد یک گره (backtracking)
 - کنش‌های برگشت‌پذیر
- جستجوی موضعی برخط
 - جستجوی موضعی عملاً یک جستجوی برخط است
 - کامل نیست ← در جستجوی برخط شروع‌های مجدد تصادفی از حالت‌های دیگر امکان ندارد
 - استفاده از گام‌برداری تصادفی
 - اگر فضا محدود باشد کامل است و کل فضای حالت را کاوش می‌کند
 - بسیار کند است

جستجوی برخط

• مثال: یک مسأله مشکل برای گام برداری تصادفی



• جستجوی موضعی برخط با استفاده از حافظه: جستجوی یادگیرنده
بلادرنگ A^* (learning real-time A^* - LRTA*)

• علاوه بر تکمیل مدل انتقال، تخمین هزینه (تابع اکتشافی پذیرفتنی) هر حالت را در
حین جستجو بهبود می دهد

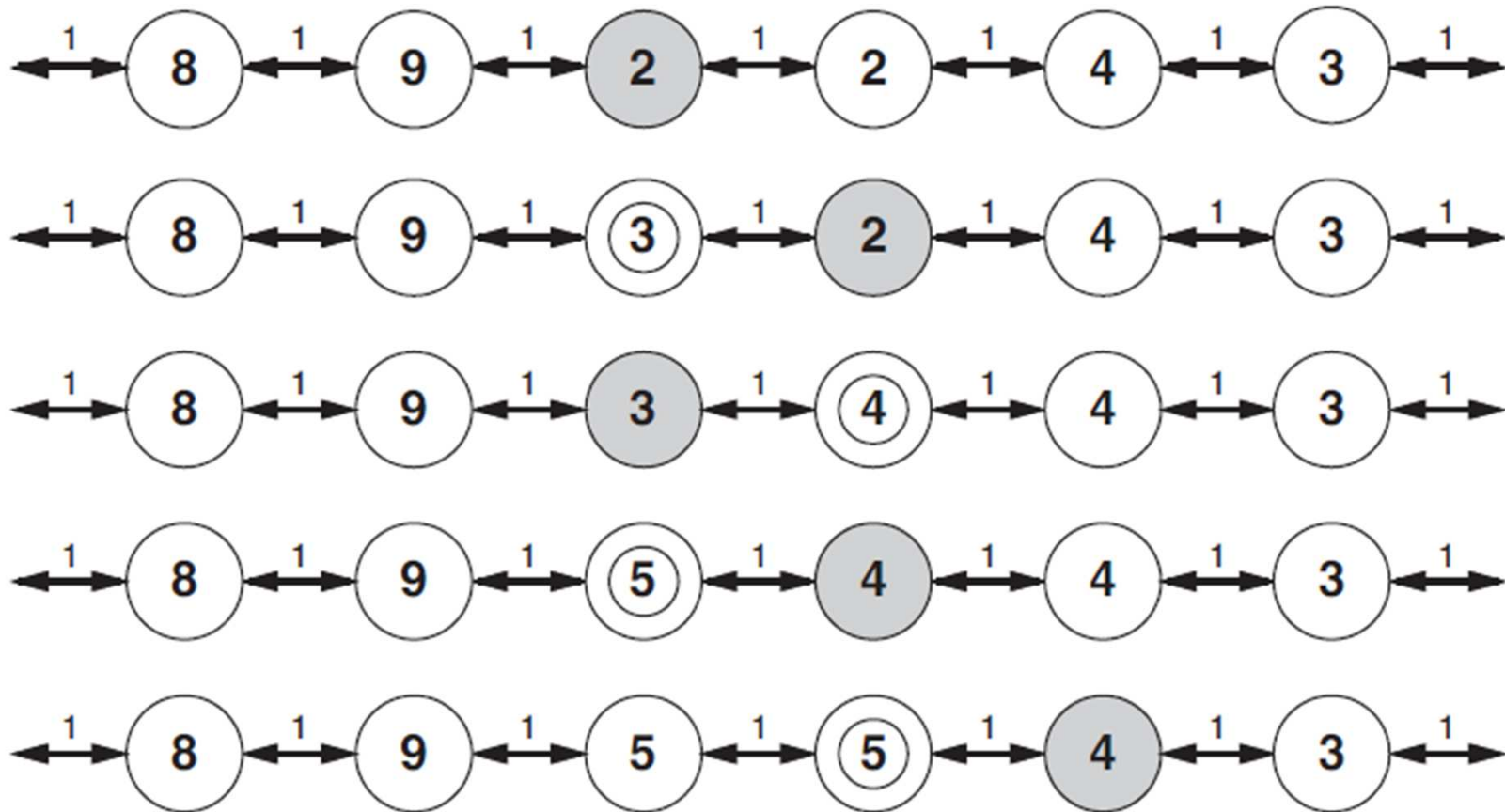
• امکان فرار از بهینه های موضعی با پر کردن حفره ها یا صاف کردن قله ها

• کنش های بررسی نشده همیشه شانس بیشتری برای انتخاب دارند ($H(s) \geq h(s)$)

• خوش بینی در هنگام تردید (optimism under uncertainty)

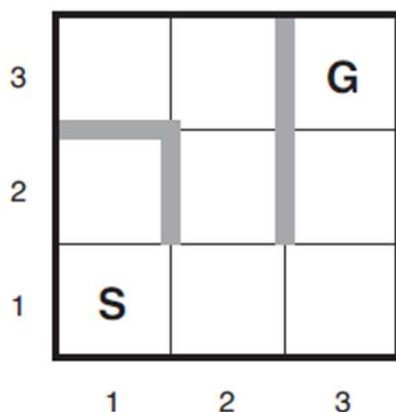
جستجوی بر خط

- جستجوی LRTA* در یک فضای حالت یک بعدی



جستجوی بر خط

- یادگیری در جستجوی بر خط
 - یادگیری نقشه محیط: مدل انتقال
 - یادگیری برآورد بهتر برای هزینه‌ی هر حالت
 - یادگیری مشخصات محیط ← استنتاج از روی حالت‌های مشاهده شده
 - بیان رسمی و قابل کار دانش دارد
 - الگوریتم‌های استنتاج قوانین کلی از روی مشاهدات خاص
 - الگوریتم‌های یادگیری ماشین



جمع‌بندی

- جستجوی موضعی برای محیط‌های پیچیده‌تر که نیاز به ذخیره مسیر طی شده ندارند
- تپه‌نوردی، تبرید شبیه‌سازی شده، جستجوی پرتوی موضعی، الگوریتم‌های ژنتیکی
- استفاده از درخت AND-OR در محیط‌هایی با کنش‌های غیرقطعی
- استفاده از حالت‌های باور در محیط‌های مشاهده‌پذیر جزئی
- استفاده از جستجوی برخط در محیط‌های ناشناخته
- مسائل کاوش