بسمه تعالی

هوش مصنوعی

جستجو در محیطهای پیچیده-۱

نیمسال اوّل ۱۴۰۳-۱۴۰۲

دکتر مازیار پالهنگ

آزمایشگاه هوش مصنوعی

دانشکدهٔ مهندسی برق و کامپیوتر

دانشگاه صنعتی اصفهان

مقدمه

- مسائلی که تاکنون در نظر می گرفتیم در محیطهای کاملاً مشاهده پذیر، قطعی، ایستا و شناخته شده بودند.
- پاسخ در این حالت دنباله ای از اعمال بود.
- در این قسمت مقداری این محدودیتها را کمتر می کنیم.
- ابتدا مسائلی که مسیر رسیدن به هدف مهم نیست.
- سپس حذف شرط قطعی بودن و کاملاً مشاهده پذیر بودن
- سپس در مورد ناشناخته بودن

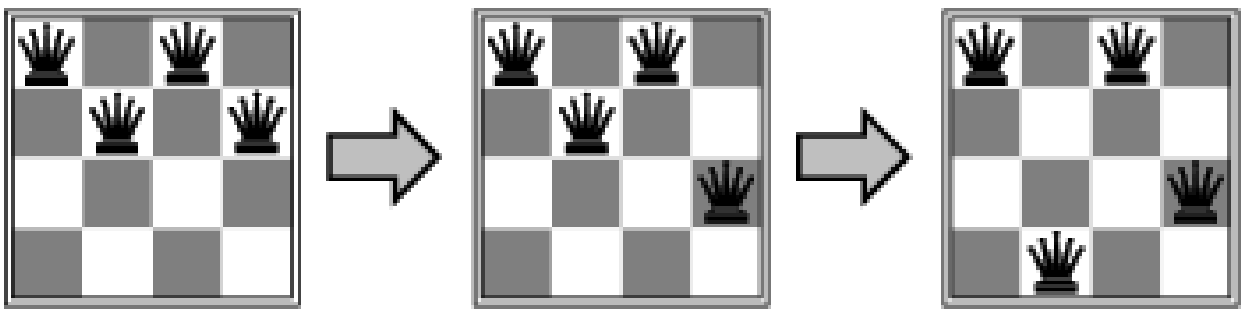
الگوریتمهای جستجوی محلی

- برخی مسائل هستند که چگونگی رسیدن به هدف در آنها مهم نیست بلکه پاسخ همان حالت هدف است.
- همانند ۸ وزیر (فقط می خواهیم چگونگی قرار گرفتن وزیرها را بدانیم نه اینکه چگونه آنرا از ابتدا پیدا کردیم.)
- فضای حالت شامل همهٔ پیکربندیهای کامل
- در این وضعیت جستجو های محلی مفید هستند.

الگوریتمهای جستجوی محلی

- حالت فعلی را نگاهدار – سعی کن آن را بهبود دهی
- مزیت این روشها:
  - معمولاً حافظهٔ کمتری مصرف می کنند،
  - اغلب در فضاهای حالت بزرگ (یا بی نهایت) جواب معقولی را می یابند که روشهای سیستماتیک توان آن را ندارند.

مثال – n وزیر



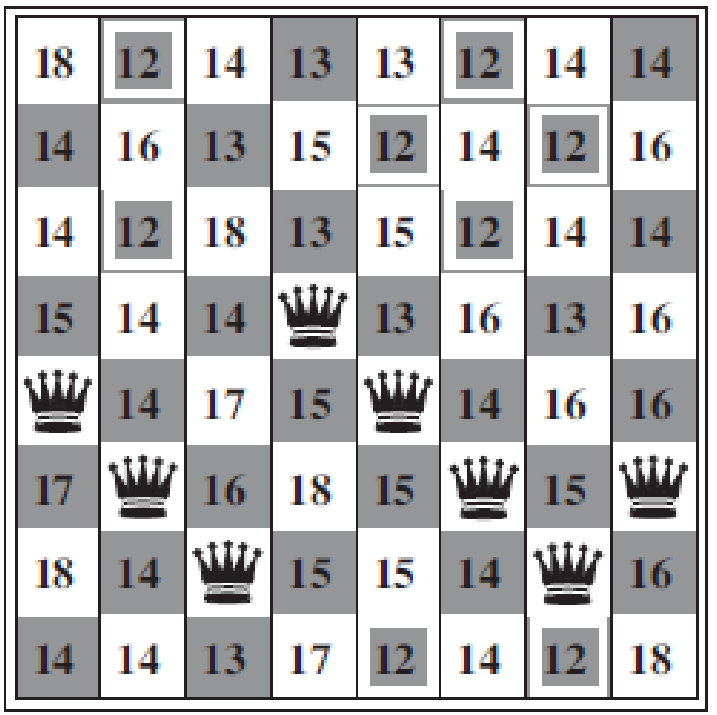
جستجوی تپه نوردی

Figure 4.2

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    current ← problem.INITIAL
    while true do
        neighbor ← a highest-valued successor state of current
        if VALUE(neighbor) ≤ VALUE(current) then return current
        current ← neighbor
```

The hill-climbing search algorithm, which is the most basic local search technique. At each step the current node is replaced by the best neighbor.

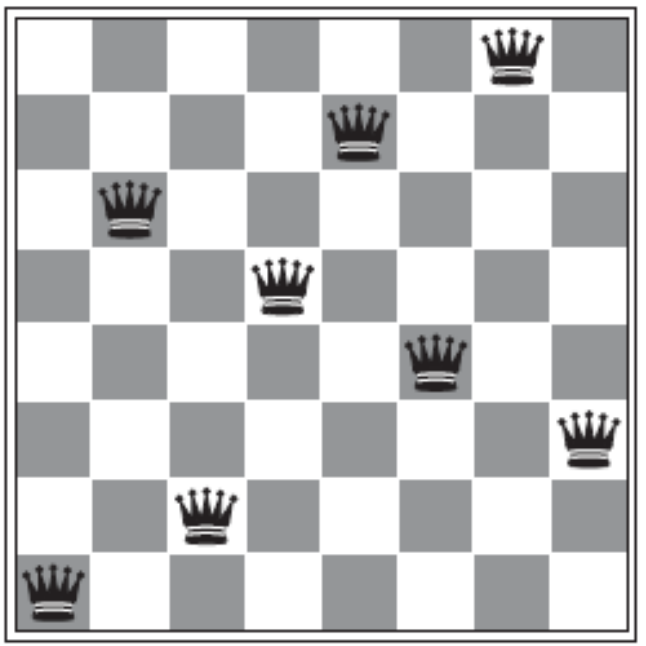
مثال – تپه نوردی (۸ وزیر)



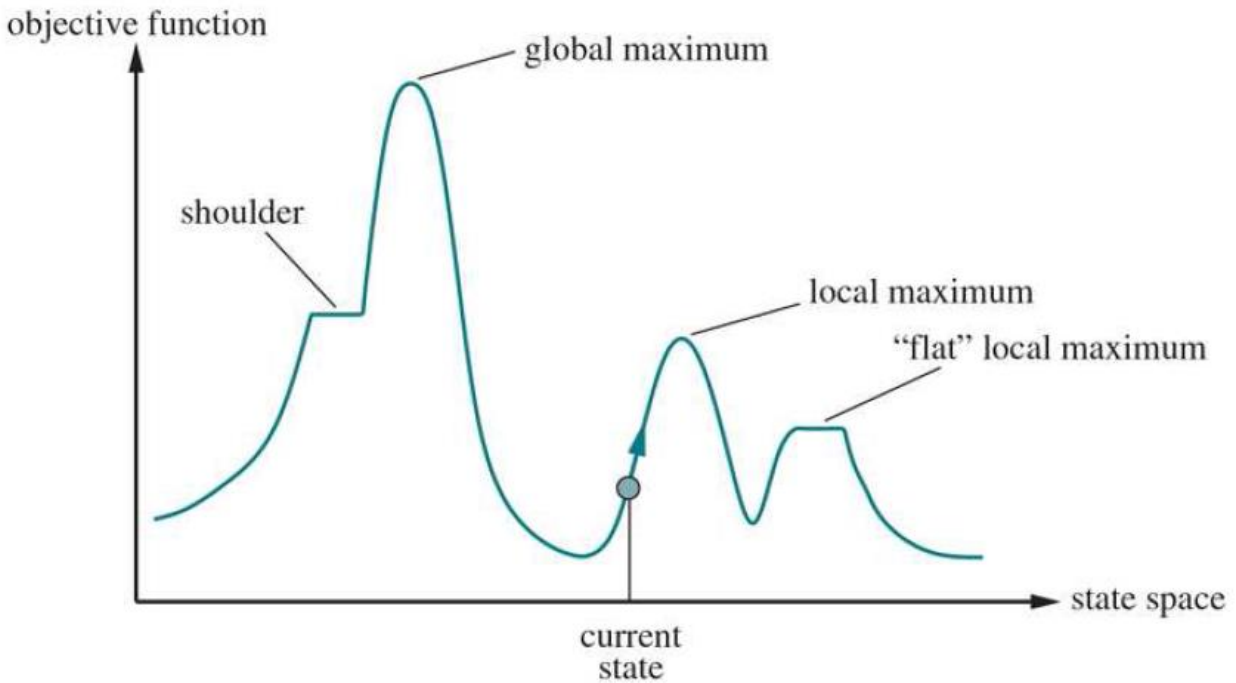
- h تعداد جفت وزیرهای که همدیگر را تهدید می کنند (مستقیم یا غیرمستقیم).
- حالت فعلی h=17
- تعداد حالات بعد ۸×۷
- بهترین حرکات علامت زده شده
- برای حالت فوق h=12

7

- بهینهٔ محلی $h=1$ هر حالت دیگر مقدار بیشتری دارد.



تپه نوردی



A one-dimensional state-space landscape in which elevation corresponds to the objective function. The aim is to find the global maximum.

تپه نوردی

- همانند اینکه با چشم بسته بخواهید از تپه یا کوهی بالا بروید.
- برای ۸ وزیر با شروع از حالت تصادفی ۸۶٪ مواقع در بهینه محلی قرار می گیرد.
- ۱۴٪ موفقیت
- ولی سریع بطور متوسط با ۴ حرکت به پاسخ صحیح، ۳ حرکت به بهینه محلی
- نسبتاً خوب برای مسئله ای با 8^8 یا حدود ۱۷ میلیون حالت.

- هنگام گیر کردن در فلات بد نیست تعداد محدودی حرکت کناری انجام گیرد به امید رهائی.
- باعث می شود از ۱۴٪ به ۹۴٪ موفقیت برای ۸ وزیر
- البته با تعداد گامهای بیشتر
- تپه نوردی گاهی جستجوی محلی حریصانه نیز نامیده می شود.
- چون بهترین حالت همسایه را بدون نگاه به آینده انتخاب می کند.

تنوعهای تپه نوردی

- در تپه نوردی تصادفی (stochastic hill climbing) انتخاب تصادفی از بین حرکات رو به بالای تپه
- احتمال انتخاب متناسب با شیب (خوبی حالت)
- معمولاً کندتر از روش تندترین صعود ولی گاهی پاسخ بهتر
- در تپه نوردی اولین انتخاب (first choice hill climbing)، تولید تصادفی حالات همسایه و انتخاب اولین حالتی که بهتر از حالت فعلی باشد.
- مناسب برای هنگامی که یک حالت تالیهای زیادی دارد.

تنوعهای تپه نوردی

- تپه نوردی با بازشروع تصادفی (random-restart hill climbing)، در صورت شکست تکرار تپه نوردی با شروع از یک حالت تصادفی دیگر.
- در صورتی که احتمال موفقیت یک تپه نوردی p باشد، تعداد تکرار مورد نیاز حدود $1/p$ می باشد.
- برای ۸ وزیر با احتمال موفقیت ۱۴٪ تعداد تکرار الگوریتم تقریباً ۷ تکرار می باشد.
- موفقیت تپه نوردی وابسته به چشم انداز است.

سرد شدن شبیه سازی شده

- تپه نوردی همواره در نظر می گیرد که حالت بعدی بهتر از حالت فعلی باشد.
- گاهی شاید بد نباشد حالت بعدی بدتر را هم بپذیریم،
- به امید اینکه در آینده به پاسخ بهتری برسیم.

سرد شدن شبیه سازی شده

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
    current ← problem.INITIAL
    for t = 1 to ∞ do
        T ← schedule(t)
        if T = 0 then return current
        next ← a randomly selected successor of current
        ΔE ← VALUE(current) – VALUE(next)
        if ΔE > 0 then current ← next
        else current ← next only with probability e^{−ΔE/T}
```

$$\frac{1}{e^{\frac{|\Delta E|}{T}}}$$

دقت شود که دراینجا ΔE منفی است

Figure 4.5 The simulated annealing algorithm, a version of stochastic hill climbing where some downhill moves are allowed. Downhill moves are accepted readily early in the annealing schedule and then less often as time goes on. The *schedule* input determines the value of the temperature T as a function of time.

مثال

- یافتن کوتاهتری مسیری که از تمام مراکز استانهای ایران عبور کند و
- از یک مرکز استان شروع و به آن مرکز استان ختم شود.
- استفاده از سرد شدن شبیه سازی شده


دریای خزر
اردبیل
تبریز
رشت
زنجان
قزوین
تهران
سنندج
کرج
همدان
کرمانشاه
قم
اراک
ایلام
خرم آباد
اصفهان
شهرکرد
اهواز
یاسوج
شیراز
بوشهر
عسلویه
خلیج فارس
گرگان
ساری
سمنان
بجنورد
مشهد
بیرجند
یزد
کرمان
زاهدان
بندرعباس
قشم
لاوان
کیش
تنب بزرگ
سیری
ابوموسی
دریای عمان

17

جستجوی پرتوی محلی

- در نظر گرفتن k حالت بجای یکی
- شروع با k حالت تصادفی تولید شده
- در هر مرحله تمامی تالیهای k حالت ایجاد می شوند
- اگر یکی از آنها هدف است توقف می شود، در غیر اینصورت k تا از بهترین تالیها انتخاب شده و کار تکرار می شود.

جستجوی پرتوی محلی

- ممکن است شبیه به اجرای موازی k بازشروع تصادفی تپه نوردی به نظر بیاید.
- تفاوت اینکه k بهترین تالی بعدی ممکن است فقط توسط برخی از حالات تولید شود.
- نهایتاً برخی از حالات دیگران را دعوت می کنند که آنها به سمت آنان برای جستجو بیایند.
- ممکن است چندان سودمند نباشد.
- چون برخی حالات که فعلاً خوب نیستند شاید بعداً پاسخهای بهتری داشته باشند.

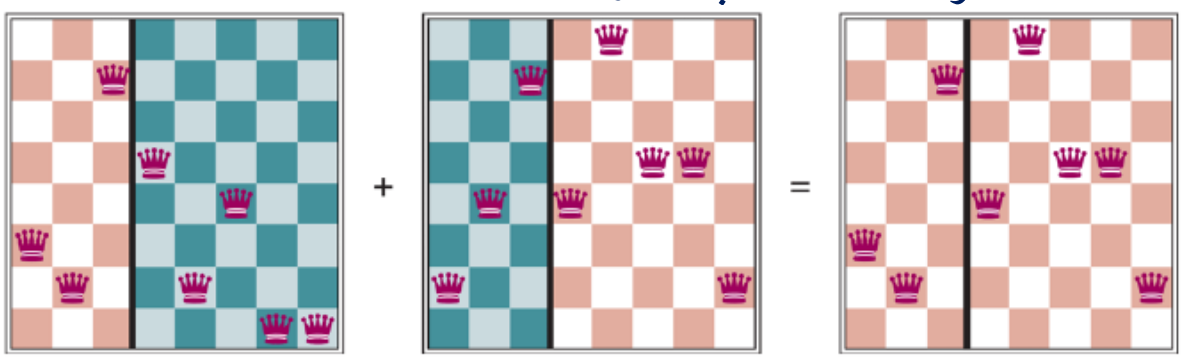
جستجوی پرتوی محلی

- راه دیگر استفاده از جستجوی پرتوی تصادفی (stochastic beam search)
- انتخاب k بهترین تالی بصورت تصادفی با احتمالی متناسب با خوبی حالت تالی.

الگوریتم ژنتیک

- شروع با k حالت تصادفی ایجاد شده (جمعیت اولیه)
- ایجاد حالات جدید با انتخاب و ترکیب اعضای موجود بر اساس یک تابع ارزیابی
- شبیه به جستجوی پرتوی تصادفی با تفاوت در روش ایجاد حالات تالی
- معمولاً نمایش حالات با رشته

- برازندگی برابر با تعداد زوج وزیرهائی که همدیگر را تهدید نمی کنند (حداکثر ۲۸ – انتخاب ۲ از ۸)



- دو حالت اول در شکل c

```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights ← WEIGHTED-BY(population, fitness)
    population2 ← empty list
    for i = 1 to SIZE(population) do
      parent1, parent2 ← WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child ← REPRODUCE(parent1, parent2)
      if (small random probability) then child ← MUTATE(child)
      add child to population2
    population ← population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n ← LENGTH(parent1)
  c ← random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```

A genetic algorithm. Within the function, *population* is an ordered list of individuals, *weights* is a list of corresponding fitness values for each individual, and *fitness* is a function to compute these values.

- در این نسخه از الگوریتم ژنتیک بجای تولید دو فرزند از والدین، فقط یک فرزند تولید شده است.

خلاصه

- حذف برخی قیود روشهای کلاسیک جستجو
- الگوریتم تپه نوردی
- الگوریتم سردشدن شبیه سازی شده
- الگوریتم پرتو محلی
- الگوریتم ژنتیک


پایان

- 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.
- حضور فعال در کلاس دارای امتیاز است.