

جستجوی موضعی

- کاوش نظام‌مند فضای جستجو در الگوریتم‌های جستجوی کلاسیک
 - در نظر گرفتن یک درخت جستجو
 - مسیرهای مختلف جستجو به صورت همزمان بررسی می‌شود
 - نگهداری گزینه‌های دیگر کاوش در هنگام بررسی یک مسیر (مجموعه مقدم)
 - یک مسیر به هدف (راه‌حل) جستجو می‌شود
- در بسیاری از مسائل مسیر رسیدن به هدف مهم نیست
 - مثال: مسأله ۸ وزیر شطرنج
 - سعی در یافتن هدف بجای یافتن مسیر رسیدن به هدف
 - نیازی به نگهداری مسیرهای مختلف (درخت جستجو) نیست



جستجوی موضعی

- در نظر گرفتن حالت فعلی در جستجوی موضعی
- عدم نیاز به نگهداری حالت‌های آماده بررسی و کاوش شده
- حالت بعدی از بین مجاورین (neighbors) حالت فعلی انتخاب می‌شود



- نیاز به حافظه بسیار کم (معمولاً ثابت)
- امکان بکارگیری در مسائلی با فضای حالت بسیار بزرگ یا نامحدود
- هدف یافتن جواب قابل قبول (reasonable) است
- لزوماً بهترین جواب (حالت هدف) را پیدا نمی‌کند

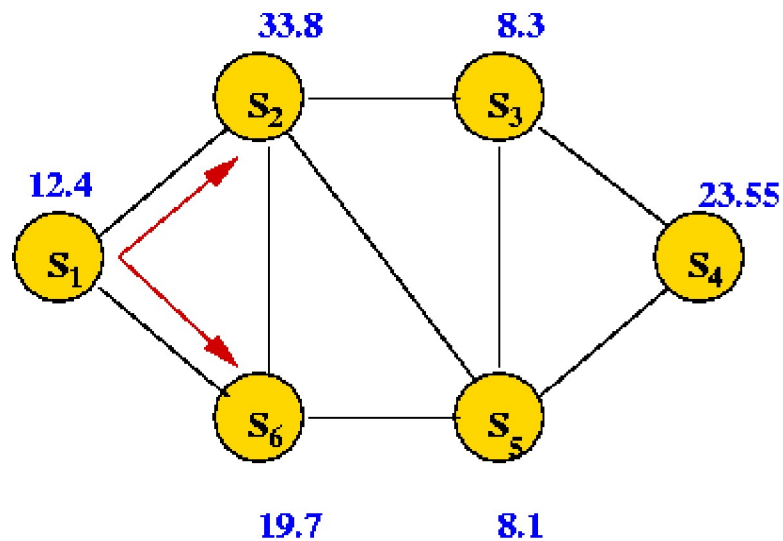
جستجوی موضعی



- تعریف مسأله برای جستجوی موضعی
 - نیاز به حالت‌ها، کنش‌ها و مدل انتقال
 - نیازی به هزینه مسیر نیست
 - مسیرها نگهداری نمی‌شوند
- استفاده از تابع هدف (objective) بجای آزمایش هدف
 - ارزش (evaluation) هر حالت از مسأله را تعیین می‌کند
- الگوریتم جستجو باید حالتی با بهترین ارزش را پیدا کند
 - یک مسأله بهینه‌سازی (optimization)
 - هدف یافتن نقطه بهینه (optimum) تابع هدف است

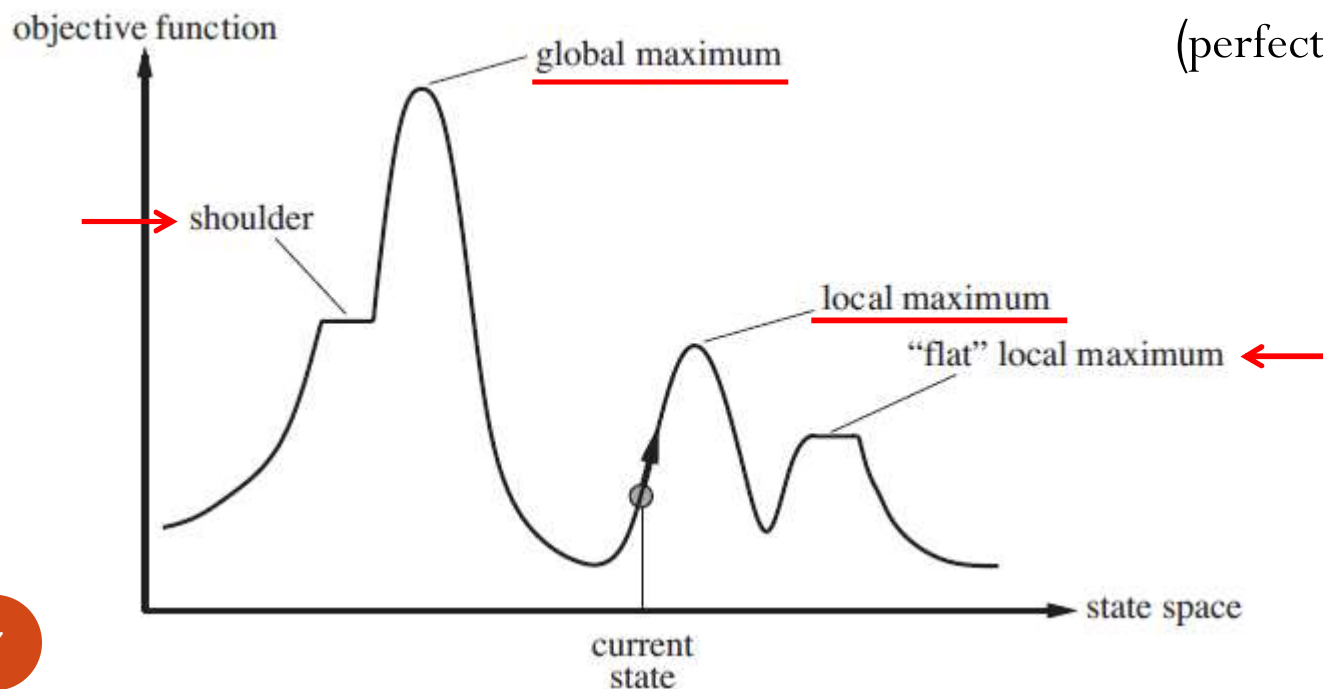
جستجوی موضعی

- چشم‌انداز (landscape) فضای حالت مسأله
- حالت‌های مسأله در مجاورت هم و ارزش آنها را نشان می‌دهد



جستجوی موضعی

- جستجوی موضعی در چشم‌انداز فضای حالت مسأله
- حالت‌های بهینه موضعی (local) و سراسری (global)
- مفاهیم کمال و بهینگی
- برخی مواقع یافتن حالت‌های بهینه موضعی نیز کافی است
- جواب عالی (perfect)



جستجوی تپه‌نوردی (hill-climbing)

- هر بار بررسی تمام حالات مجاور حالت فعلی و انتخاب بهترین آنها
- حالات مجاور با یک کنش از حالت فعلی بدست می‌آیند
- حرکت در جهت افزایش (کاهش) مقدار تابع هدف
- تندترین صعود (steepest ascent) یا شیب نزولی (gradient descent) در بهینه‌سازی
- توقف با رسیدن به یک قله (دره)

```
function HILL-CLIMBING(problem) returns a state that is a local maximum  
  
  current  $\leftarrow$  MAKE-NODE(problem.INITIAL-STATE)  
  loop do  
    neighbor  $\leftarrow$  a highest-valued successor of current  
    if neighbor.VALUE  $\leq$  current.VALUE then return current.STATE  
    current  $\leftarrow$  neighbor
```

• جستجوی موضعی حریصانه

• حالات فراتر از حالات مجاور را در نظر نمی‌گیرد

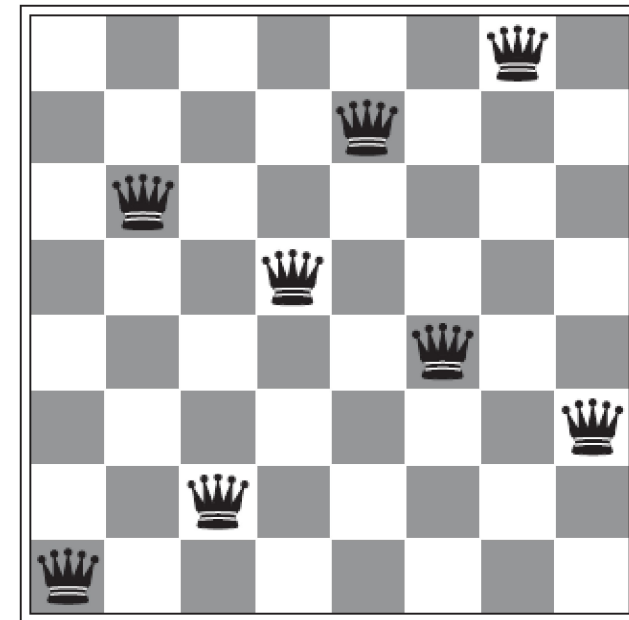
جستجوی تپه‌نوردی

- مثال: ۸ وزیر شطرنج (با تعریف حالت کامل)
- تابع هدف (h): تعداد وزیرانی که در هر حالت یکدیگر را تهدید می‌کنند
- هر حالت دارای ۵۶ حالت مجاور است

$h = 17$

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♔	13	16	13	16
♔	14	17	15	♔	14	16	16
17	♔	16	18	15	♔	15	♔
18	14	♔	15	15	14	♔	16
14	14	13	17	12	14	12	18

$h = 1$



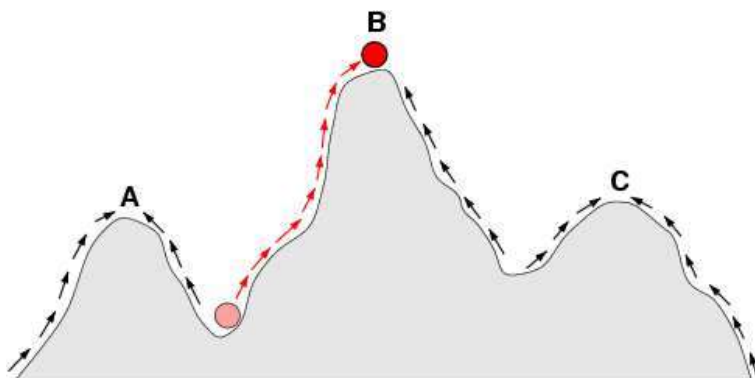
در
قدم

جستجوی تپه‌نوردی

- الگوریتم می‌تواند گیر کند (gets stuck)

- بهینه‌های موضعی

- بستگی به نقطه شروع جستجو



- لبه‌ها – شیارها (ridges)

- دنباله‌ای از بهینه‌های موضعی که

- مستقیماً به هم مرتبط نیستند

- سطوح مسطح (plateau)

- بهینه‌های موضعی مسطح یا شانه‌ها

- استفاده از حرکتهای به کنار (sideways move)

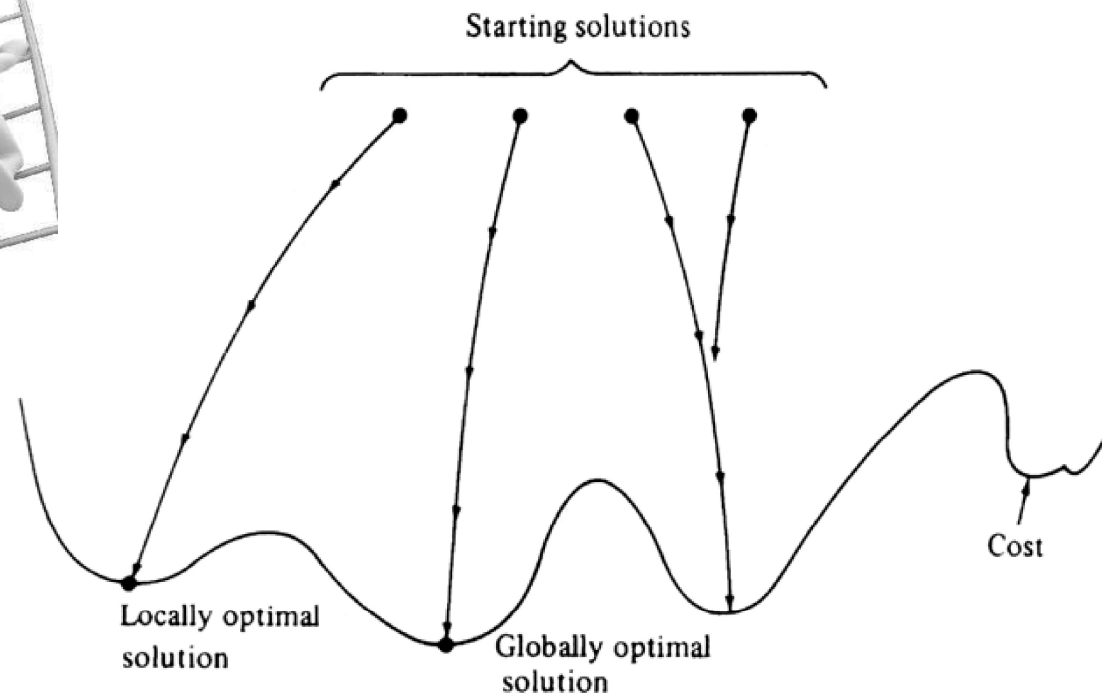
- در نظر گرفتن یک بیشینه و بخاطر سپاری حالت‌های کاوش شده برابر

جستجوی تپه‌نوردی

- گونه‌های دیگر الگوریتم
 - تپه‌نوردی تصادفی (stochastic)
 - انتخاب تصادفی از بین مجاورین بهتر
 - احتمال حالت بعدی بسته به تندی شیب رسیدن به آن دارد
 - تپه‌نوردی اولین گزینه (first-choice)
 - تولید تعدادی حالت مجاور بصورت تصادفی تا یافتن یک مجاور بهتر
 - مناسب برای مسائلی که هر حالت دارای تعداد بسیار زیادی حالت مجاور است
 - تپه‌نوردی با شروع مجدد تصادفی (random-restart)
 - بکارگیری یک دنباله از تپه‌نوردی‌ها با شروع از حالت‌های اولیه‌ی تصادفی
 - بهترین حالت یافته شده در میان تمام تپه‌نوردی‌ها جواب است

جستجوی تپه‌نوردی

- تپه‌نوردی با شروع مجدد تصادفی



- کارایی الگوریتم‌های مختلف بسته به وضعیت چشم‌انداز مسأله دارد
- تعداد بهینه‌های موضعی

جستجوی تپهنوردی

- ویژگی‌های الگوریتم
- تقریباً هیچ گونه‌ای از الگوریتم کامل نیست
 - در بهینه‌های محلی گیر می‌کند
- احتمال کامل بودن تپهنوردی با شروع مجدد تصادفی با افزایش تکرارها به یک میل می‌کند
- نهایتاً یکی از حالت‌های شروع تصادفی حالت هدف خواهد بود
- اگر p احتمال موفقیت هر تپهنوردی باشد، تعداد شروع‌های مجدد لازم $1/p$ خواهد بود
- هزینه جستجو بر حسب تعداد گام‌های مورد نیاز برای یافتن هدف

متوسط گام‌ها
در تکرارهای موفق

$$N_s + \frac{1-p}{p} N_f$$

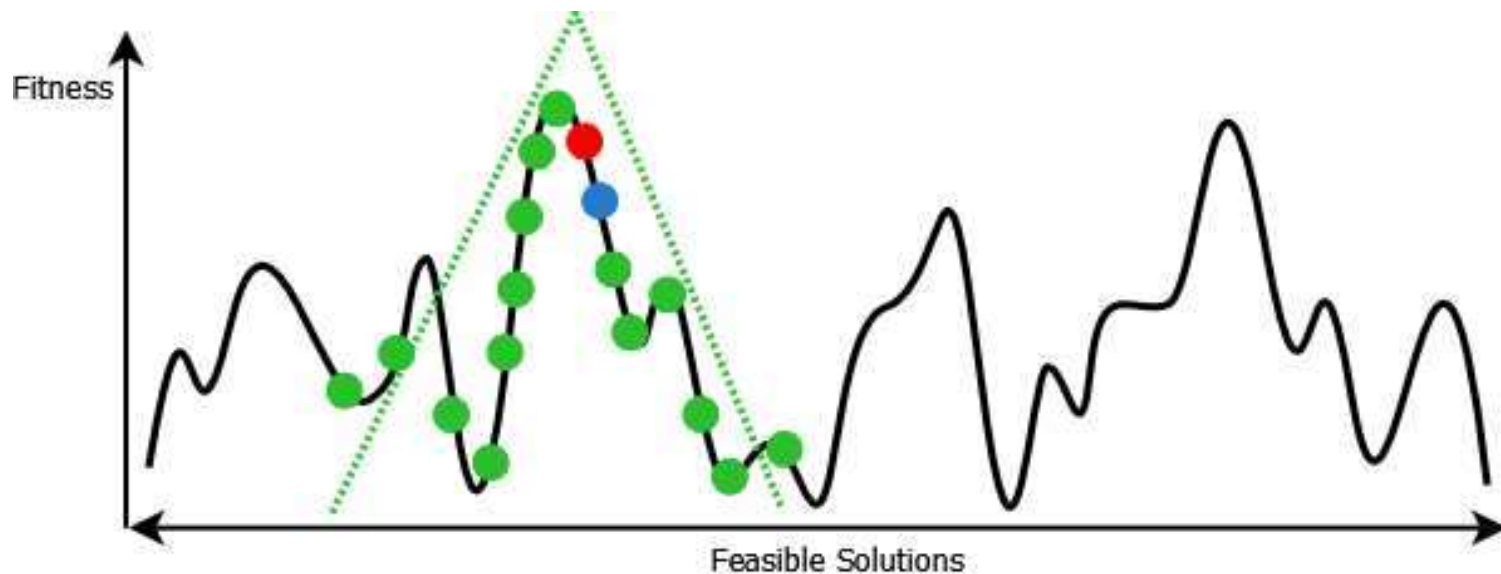
متوسط گام‌ها
در تکرارهای ناموفق

تبرید شبیه‌سازی شده (Simulated Annealing)

- جستجوی گام‌برداری تصادفی (random walk)
 - از بین مجاورین یک حالت بصورت تصادفی با توزیع یکنواخت انتخاب می‌شود
 - امکان انتخاب حالت‌های مجاور بدتر
 - کامل است اما کارایی بسیار ضعیفی دارد
- جستجوی تپه‌نوردی کارایی بالایی دارد اما کامل نیست
 - همیشه حالت‌های مجاور بهتر را به صورت حریصانه انتخاب می‌کند
- سعی در تلفیق این دو راه‌برد
 - دستیابی به کمال و کارایی بصورت همزمان
 - جستجوی تبرید شبیه‌سازی شده با این رویکرد طراحی شده است

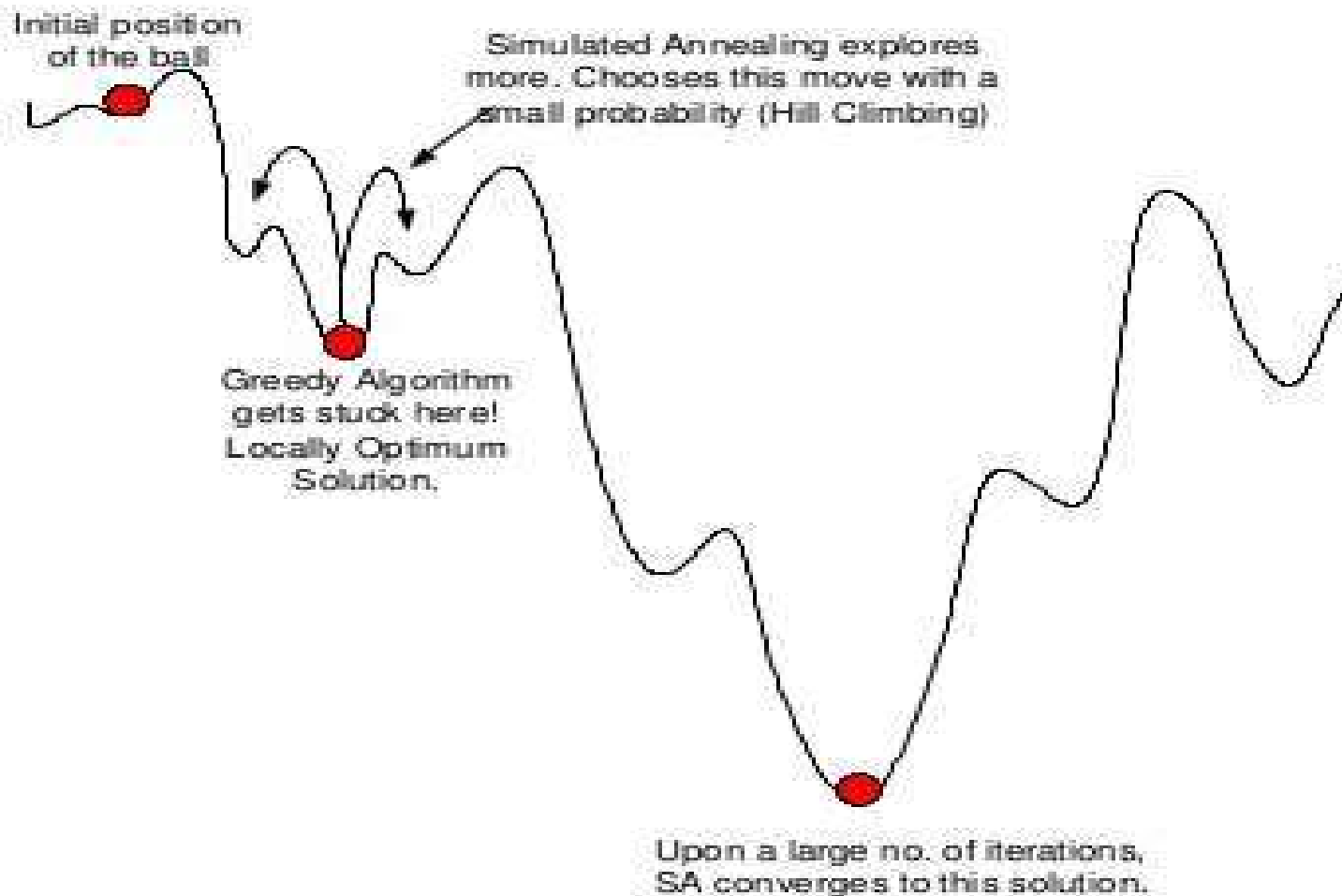
تبرید شبیه‌سازی شده

- هدف تبرید: کمینه کردن انرژی جنبشی
- آبدیده کردن فلزات با گرم کردن و سپس سرد کردن تدریجی
- امکان فرار از بهینه‌های موضعی
- قبول حرکت به حالت‌های بدتر و کاهش تدریجی بسامد این حرکات



تبرید شبیه‌سازی شده

- شبیه حرکت یک توپ بر روی یک صفحه نامسطح با کمک لرزاندن



تبرید شبیه‌سازی شده

- الگوریتم جستجوی تبرید شبیه‌سازی شده

function SIMULATED-ANNEALING(*problem*, *schedule*) **returns** a solution state

inputs: *problem*, a problem

schedule, a mapping from time to “temperature”

current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

for $t = 1$ **to** ∞ **do**

$T \leftarrow \text{schedule}(t)$

← تعیین دمای فعلی

if $T = 0$ **then return** *current*

next \leftarrow a randomly selected successor of *current*

$\Delta E \leftarrow \text{next.VALUE} - \text{current.VALUE}$

if $\Delta E > 0$ **then** *current* \leftarrow *next*

else *current* \leftarrow *next* only with probability $e^{\Delta E/T}$

$$P(\text{bad move}) = \frac{1}{e^{|\Delta E|/T}}$$

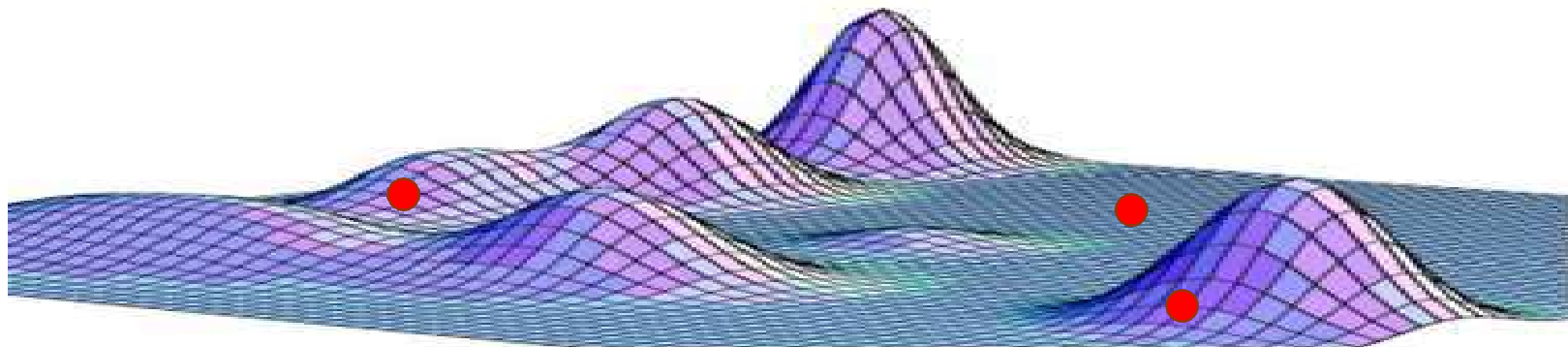
- نیاز به یک زمانبندی (*schedule*) برای کاهش تدریجی حرکات بد

- با زمانبندی مناسب احتمال کامل بودن الگوریتم به یک میل می‌کند

- زمانبندی مناسب؟

جستجوی پرتوی موضعی (local beam)

- نگهداری و بررسی همزمان چندین (k) حالت
- هر بار حالت‌های مجاور تمام حالت‌های فعلی بررسی و k همسایه بهتر برای مرحله بعدی الگوریتم انتخاب می‌شوند
- بهینه‌سازی با استفاده از یک جمعیت (population) از حالت‌های مسأله
- تبادل اطلاعات بین جستجوهای موازی در مورد حالت‌های امید بخش
- تفاوت با k تکرار موازی تپه‌نوردی با شروع مجدد تصادفی



جستجوی پرتوی موضعی

- امکان همگرایی زودرس (premature convergence)
- تمام حالت‌های فعلی در یک منطقه از فضای جستجو متمرکز می‌شوند
- عدم امکان کاوش مناسب تمام فضای جستجو
- شبیه به جستجوی تپه‌نوردی خواهد شد

- جستجوی پرتوی تصادفی (stochastic)
- بجای انتخاب k بهترین همسایه، k همسایه بصورت تصادفی انتخاب می‌شوند
- شانس انتخاب هر همسایه متناسب با ارزش آن است

الگوریتم ژنتیکی (Genetic Algorithm)

- نگهداری یک جمعیت از افراد (حالت‌های مسأله)
- ارزیابی افراد با استفاده از یک تابع برازندگی
- تابع هدف مسأله جستجو
- بکارگیری نوع خاصی از نمایش فاکتوربندی شده برای نشان دادن افراد
 - هر فرد بصورت یک رشته (کروموزوم – chromosome) از متغیرها (ژن‌ها – Genes) با دامنه مقادیر (alleles) مشخص در نظر گرفته می‌شود
 - مثلاً در رشته‌های بیتی دامنه مقادیر 0 و 1 است
- استفاده از عملگرهای ژنتیکی بجای کنش‌های مسأله
 - عملگرهای تقطیع (crossover) و جهش (mutation)

الگوریتم ژنتیکی

• مراحل الگوریتم

- تولید یک جمعیت اولیه از افراد (معمولاً به صورت تصادفی)
- ارزیابی افراد جمعیت و تعیین برازندگی آنها
- انتخاب برخی از افراد بر اساس برازندگی به عنوان والدین (parents)
- جفت‌سازی (pairing) والدین (بر اساس برازندگی آنها)
- اعمال تقطیع روی هر جفت با احتمال P_c و تولید یک جفت فرزند
- اعمال جهش روی هر یک از فرزندان با احتمال P_m
- ارزیابی کلیه فرزندان (offspring) تولید شده و تعیین برازندگی آنها
- جایگزینی (replacement) فرزندان در جمعیت افراد
- بازگشت به مرحله انتخاب والدین تا محقق شدن شرط توقف الگوریتم

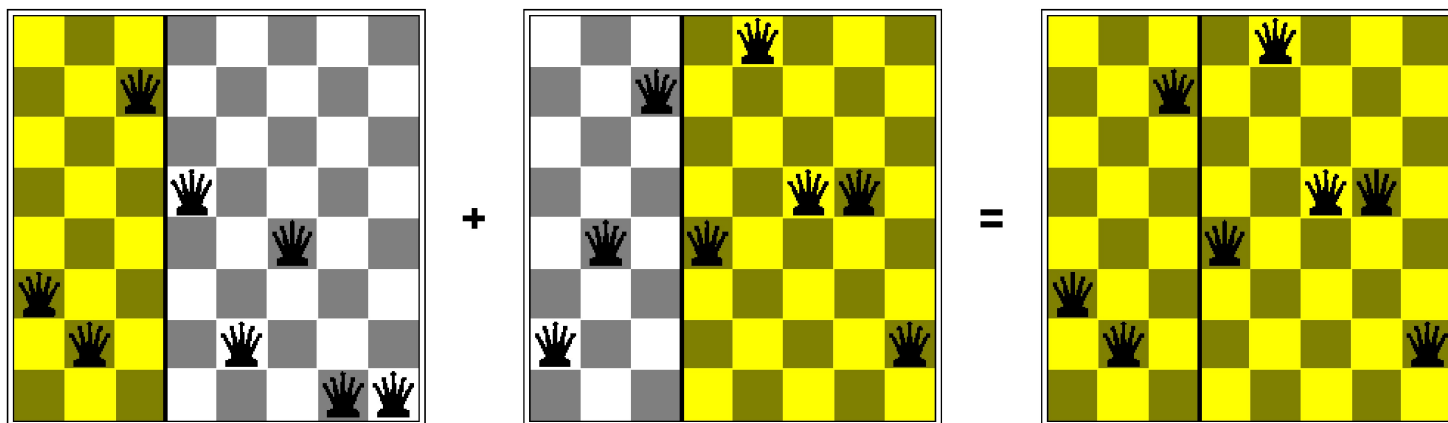
(reproduction)

تولید

یک نسل (generation)

الگوریتم ژنتیکی

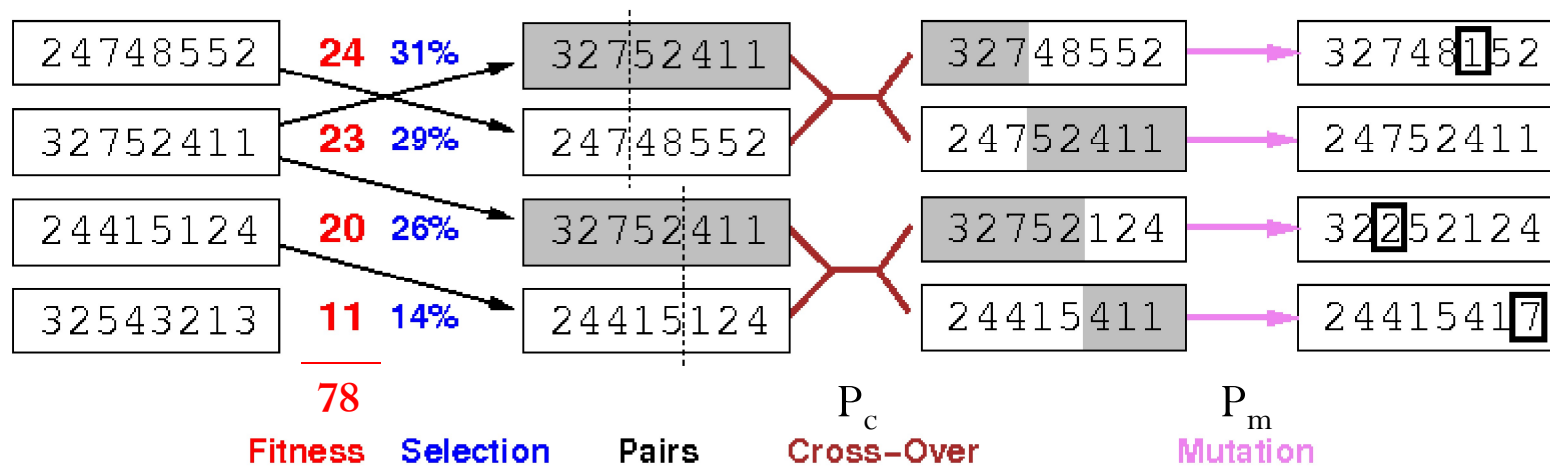
- عملگر تقطیع با توجه به نقطه (های) تقطیع در رشته
- ترکیب قسمت‌های سمت چپ و راست نقطه تقطیع از دو والد
- مثال: عملگر تقطیع در مسأله ۸ وزیر



- تقطیع باعث افزایش سطح پیش‌روی (granularity) جستجو می‌شود
- شبیه به اعمال چندین کنش در یک حالت از مسأله
- فاصله والدین به مرور (طی نسل‌ها) نسبت به هم کم می‌شود

الگوریتم ژنتیکی

- عملگر جهش در یک رشته
- مقدار هر ژن با احتمالی (P_m) تغییر می کند
- انتخاب تصادفی یکی از مقادیر ممکن جدید برای متغیر (ژن) مشخص شده
- شبیه به اعمال یک کنش در یک حالت از مسأله
- مثال: یک نسل از الگوریتم در مسأله ۸ وزیر



الگوریتم ژنتیکی

- یک الگوریتم بسیار پرکاربرد در مسائل بهینه‌سازی
- بخصوص در مسائل چندهدفه (multi-objective)
- کامل نیست
- یک الگوریتم جستجوی موضعی است
- تئوری شِما (schema theory) برای تحلیل عملکرد الگوریتم
- بررسی نحوه تکامل افراد در طی نسل‌های پی‌درپی
- پیچیدگی زمانی و فضایی محدود
- دارای نسخه‌های متنوع برای مسائل مختلف
- گونه‌ای از دسته کلی‌تر الگوریتم‌های تکاملی (evolutionary algorithms)