

- 教材讨论
 - JH第3章第6节第1、2小节

问题1: local search的基本概念

- 你能解释这些术语的含义吗?
基于此, 你能解释local search的基本思想吗?

- feasible solution
- transformation
- neighborhood
- local optimum

LSS(*Neigh*)-Local Search Scheme according to a neighborhood *Neigh*

Input: An input instance x of an optimization problem U .

Step 1: Find a feasible solution $\alpha \in \mathcal{M}(x)$.

Step 2: **while** $\alpha \notin \text{LocOPT}_U(x, \text{Neigh}_x)$ **do**

begin find a $\beta \in \text{Neigh}_x(\alpha)$ such that

$\text{cost}(\beta) < \text{cost}(\alpha)$ if U is a minimization problem and

$\text{cost}(\beta) > \text{cost}(\alpha)$ if U is a maximization problem; $\alpha := \beta$

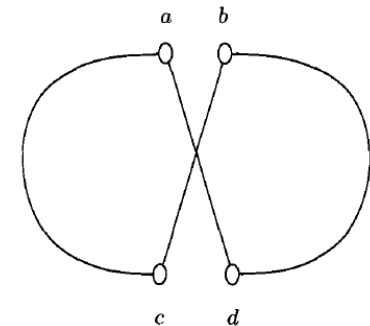
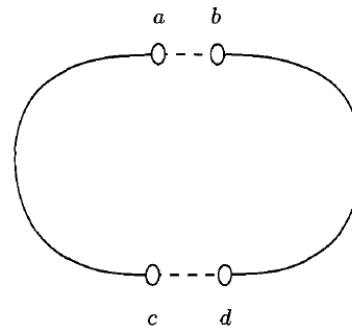
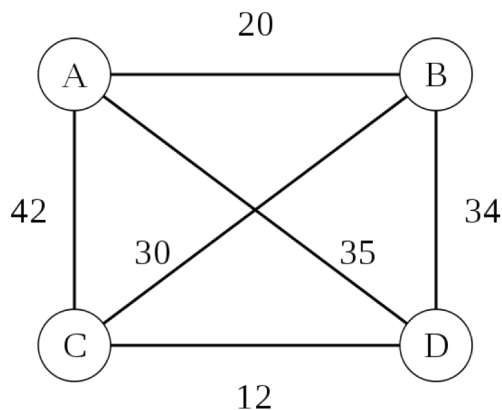
end

Output: **output**(α).

- 你能证明LSS的total correctness吗?
- 决定LSS能否并尽快找到全局最优解的因素有哪些?
 - α
 - *Neigh*
 - β

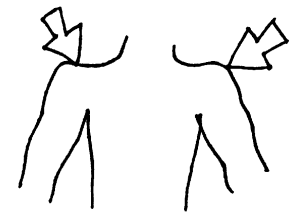
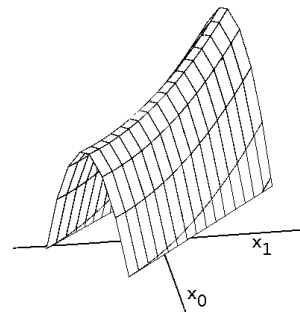
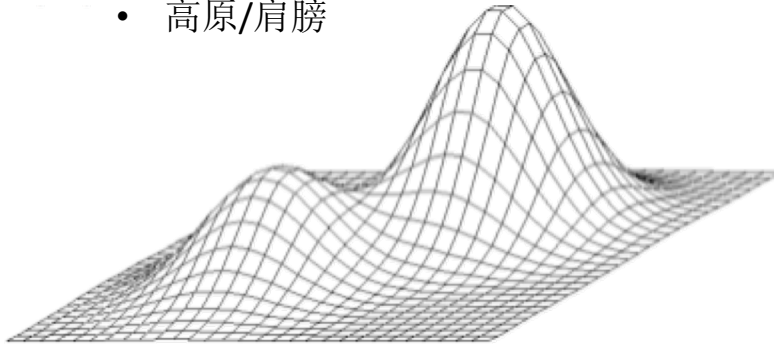
问题2: hill climbing

- In computer science, hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.
 - 这里的 α 、Neigh、 β 分别是怎么取的？
 - 你能以TSP为例，给出一个具体的算法吗？

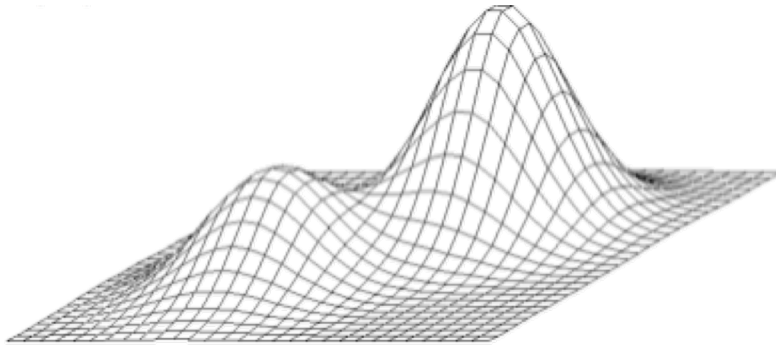


问题2: hill climbing (续)

- In computer science, hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.
 - 你认为hill climbing存在哪些问题?
 - 局部最优
 - 缓升（如：之字形爬升非轴向的山脊）
 - 高原/肩膀

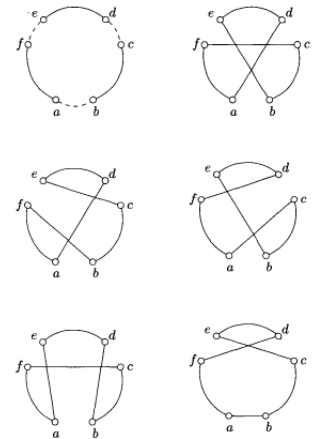
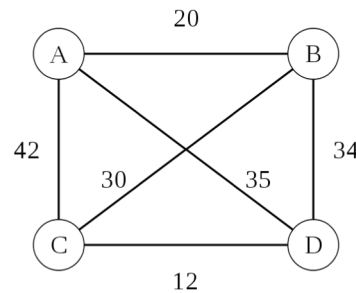
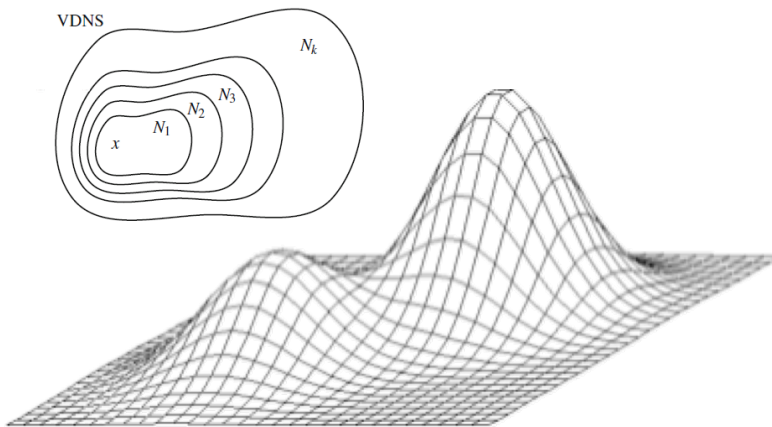


- 你能想到哪些策略来缓解这些问题?
 - α
 - Neigh
 - β



问题3: very large-scale neighborhood search

- A very large-scale neighborhood search is a local search algorithm which makes use of a neighborhood definition, which is **large and possibly exponentially sized**.
 - 和hill climbing相比, 它改变了 α 、Neigh、 β 中的哪一个? 这样做有什么好处?
 - 你能以TSP为例, 给出一个具体的算法吗?



- 你认为朴素的very large-scale neighborhood search存在什么问题?
- 你能想出折中的策略来应对这个问题吗?

问题3: very large-scale neighborhood search (续)

- Variable-depth search methods are techniques that search the k -exchange neighborhood **partially**, hence reducing the time used to search the neighborhood.
 - KL(Neigh)是如何实现“partially”的?
你能简述它的思想吗?
 - KL(Neigh)与hill climbing的区别是什么?
 - KL(Neigh)允许中途下山
 - 你能以TSP为例, 给出一个具体的算法吗?

KL(Neigh) Kernighan-Lin Variable-Depth Search Algorithm with respect to the neighborhood *Neigh*

Input: An input instance I of an optimization problem U .

Step 1: Generate a feasible solution $\alpha = (p_1, p_2, \dots, p_n) \in \mathcal{M}(I)$ where (p_1, p_2, \dots, p_n) is such a parametric representation of α that the local transformation defining *Neigh* can be viewed as an exchange of a few of these parameters.

Step 2: *IMPROVEMENT* := *TRUE*;

EXCHANGE := $\{1, 2, \dots, n\}$; $J := 0$; $\alpha_J := \alpha$;

while *IMPROVEMENT* = *TRUE* **do begin**

while *EXCHANGE* $\neq \emptyset$ **do**

begin $J := J + 1$;

$\alpha_J :=$ a solution from *Neigh*(α_{J-1}) such that *gain*(α_{J-1}, α_J) is the maximum of

$\{gain(\alpha_{J-1}, \delta) \mid \delta \in Neigh(\alpha_{J-1}) - \{\alpha_{J-1}\} \text{ and } \delta \text{ differs from } \alpha_{J-1} \text{ in the parameters of } EXCHANGE \text{ only}\}$;

EXCHANGE := *EXCHANGE* - {the parameters in which α_J and α_{J-1} differ}

end;

 Compute *gain*(α, α_i) for $i = 1, \dots, J$;

 Compute $l \in \{1, \dots, J\}$ such that

$$gain(\alpha, \alpha_l) = \max\{gain(\alpha, \alpha_i) \mid i \in \{1, 2, \dots, J\}\};$$

if *gain*(α, α_l) > 0 **then**

begin $\alpha := \alpha_l$;

EXCHANGE := $\{1, 2, \dots, n\}$

end

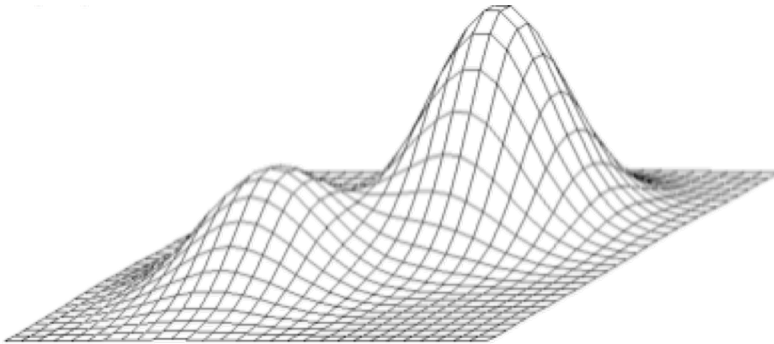
else *IMPROVEMENT* := *FALSE*

end

Step 3: **output**(α).

问题4: multi-start methods

- **Re-start** the procedure from a new solution once a region has been explored.
 - 和hill climbing相比, 它改变了 α 、Neigh、 β 中的哪一个? 这样做有什么好处?
 - 你认为应该如何选择new solution?
 - 你能以TSP为例, 给出一个具体的算法吗?



问题4: multi-start methods (续)

- Greedy Randomized Adaptive Search Procedure (GRASP)
 - The GRASP metaheuristic is a multi-start or iterative process, in which each iteration consists of two phases: construction and local search. The construction phase builds a feasible solution, whose neighborhood is investigated until a local minimum is found during the local search phase. The best overall solution is kept as the result.

```
procedure GRASP(Max_Iterations,Seed)
1  Read_Input();
2  for  $k = 1, \dots, \text{Max\_Iterations}$  do
3      Solution  $\leftarrow$  Greedy_Randomized_Construction(Seed);
4      Solution  $\leftarrow$  Local_Search(Solution);
5      Update_Solution(Solution,Best_Solution);
6  end;
7  return Best_Solution;
end GRASP.
```

FIGURE 1. Pseudo-code of the GRASP metaheuristic.

```
procedure Greedy_Randomized_Construction(Seed)
1  Solution  $\leftarrow$   $\emptyset$ ;
2  Evaluate the incremental costs of the candidate elements;
3  while Solution is not a complete solution do
4      Build the restricted candidate list (RCL);
5      Select an element  $s$  from the RCL at random;
6      Solution  $\leftarrow$  Solution  $\cup$   $\{s\}$ ;
7      Reevaluate the incremental costs;
8  end;
9  return Solution;
end Greedy_Randomized_Construction.
```

FIGURE 2. Pseudo-code of the construction phase.

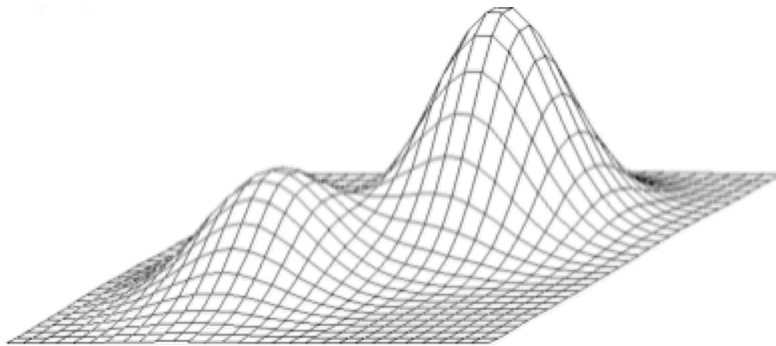
Greedy、Randomized、Adaptive
分别是如何体现的？

```
procedure Local_Search(Solution)
1  while Solution is not locally optimal do
2      Find  $s' \in N(\text{Solution})$  with  $f(s') < f(\text{Solution})$ ;
3      Solution  $\leftarrow$   $s'$ ;
4  end;
5  return Solution;
end Local_Search.
```

FIGURE 3. Pseudo-code of the local search phase.

问题5: stochastic hill climbing

- Stochastic hill climbing chooses at random from among the uphill moves.
 - 和hill climbing相比, 它改变了 α 、Neigh、 β 中的哪一个? 这样做有什么好处?
 - 你认为应该如何计算每一种移动的概率?
 - The probability of selection can vary with the steepness of the uphill move.
 - 你能以TSP为例, 给出一个具体的算法吗?



- 我们似乎已经讨论了对hill climbing的所有可能的改进策略
 - α : multi-start methods (GRASP)
 - Neigh: very large-scale neighborhood search (variable-depth search)
 - β : stochastic hill climbing
- 你还能想到别的方法吗?
 - 计算机除了“算”以外，还能做什么？

问题6: tabu search

- Tabu search enhances the performance of local searches by using **memory structures** that describe the visited solutions or user-provided sets of rules.
 - Short-term: The list of solutions recently considered. If a potential solution appears on the tabu list, it cannot be revisited until it reaches an expiration point.
 - Intermediate-term: Intensification rules intended to bias the search towards promising areas of the search space.
 - Long-term: Diversification rules that drive the search into new regions (i.e. regarding resets when the search becomes stuck in a plateau or a suboptimal dead-end).
- 你能以TSP为例，给出一个具体的算法吗？

问题7: local search的性能

LSS(*Neigh*)-Local Search Scheme according to a neighborhood *Neigh*

Input: An input instance x of an optimization problem U .

Step 1: Find a feasible solution $\alpha \in \mathcal{M}(x)$.

Step 2: **while** $\alpha \notin \text{LocOPT}_U(x, \text{Neigh}_x)$ **do**
 begin find a $\beta \in \text{Neigh}_x(\alpha)$ such that
 $\text{cost}(\beta) < \text{cost}(\alpha)$ if U is a minimization problem and
 $\text{cost}(\beta) > \text{cost}(\alpha)$ if U is a maximization problem; $\alpha := \beta$
 end

Output: **output**(α).

- local search的运算时间受哪些因素的影响?

问题8：应用

- 你能综合运用我们讨论的这些策略，分别为下列问题设计一种local search算法吗？
 - longest simple path
 - MAX-SAT
 - MAX-CL
 - MIN-VCP