Artificial Intelligence (CS303)

Lecture 2: Beyond Classical Search

Hints for this lecture

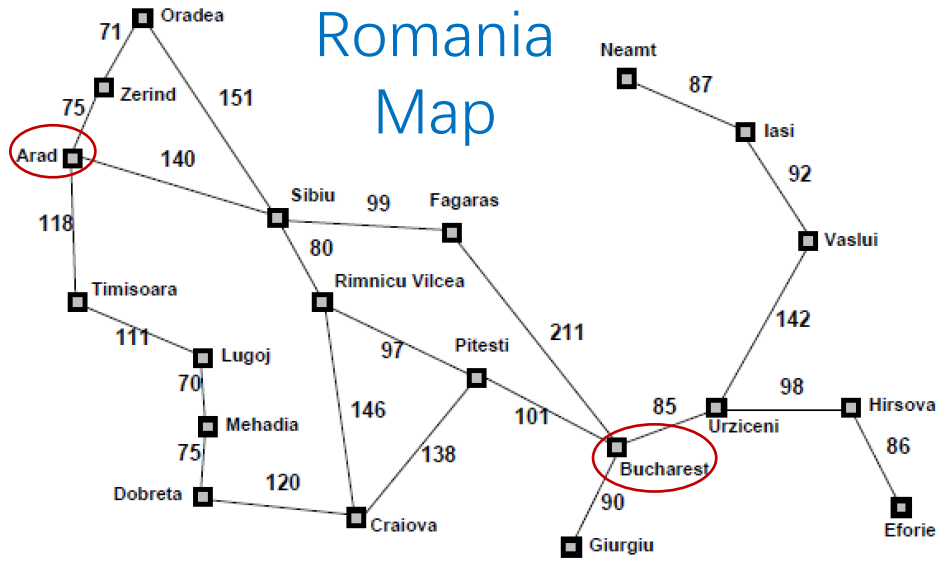
- **“Classical” search: use a tree representation**
- **“Beyond”: generalize to other types of representation**

Outline of this lecture

- More Representations
- General Search Methods
- More complex real-world search scenarios

More representations

- Route planning on Romania map again…
- In lecture 1, we formulated the problem with tree representation.
- Any different way to formulate it?
- In other words, what is a “state”?
- Alternative: permutations of cities
- Which one is better?



More representations

- As long as the state space is finite (finite set), tree representation can be used to formulate any problem.

- Any problem solved on modern computing hardware can be viewed as having a finite state space.

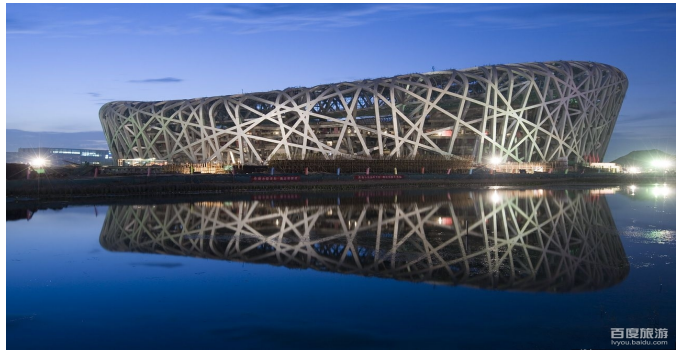
- In many cases, Search Tree is not the most appropriate representation.

More representations

- **Consider the ubiquitous optimization problems.**

Transportation: Railway timetabling

Energy: Wind turbine design



Architecture: Truss design (Birds nest)

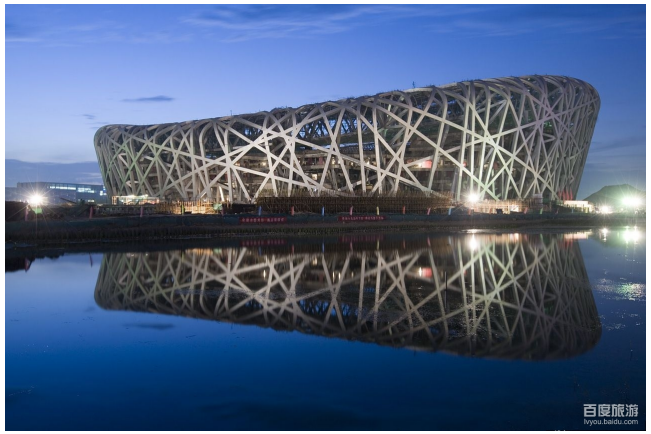
Finance: Portfolio optimization

$$\text{maximize} \quad f(x)$$

$$\text{subject to:} \quad g_i(x) \le 0, \quad i = 1 \ldots m$$

$$h_j(x) = 0, \quad j = 1 \ldots p$$

More representations



Truss Design

? ? ?

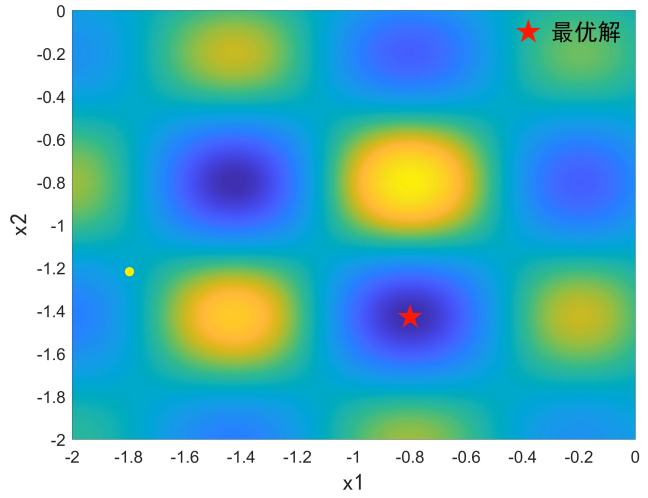
$$\begin{aligned} \text{maximize} \quad & f(x) \\ \text{subject to:} \quad & g_i(x) \le 0, \quad i = 1 \ldots m \\ & h_j(x) = 0, \quad j = 1 \ldots p \end{aligned}$$

What is the mathematical formulation of these functions?

Direct Search in the Solution Space

- When tackling an optimization problem, it is more natural to imaging searching for a solution in the solution space.

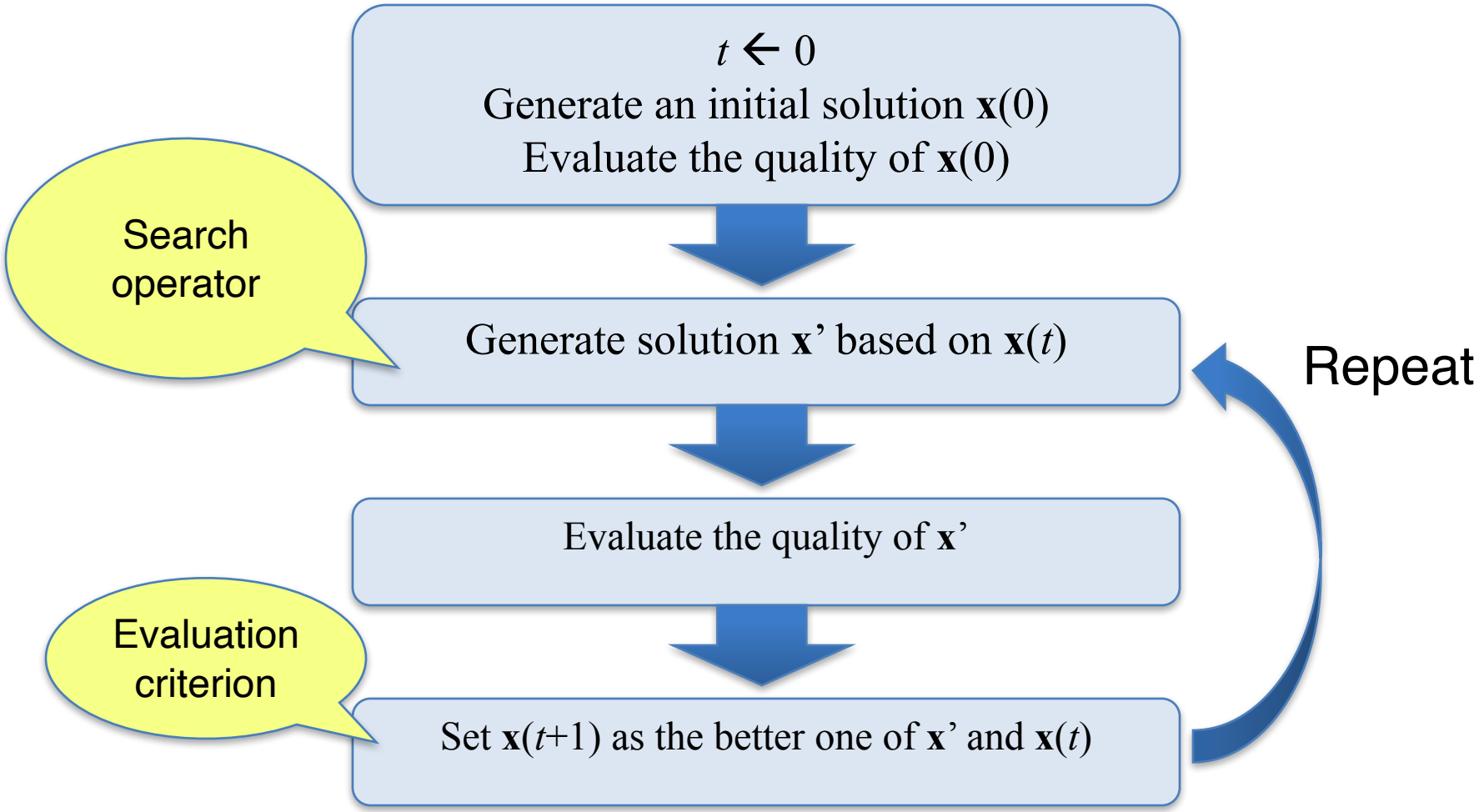
- Each element of the solution space is a complete solution.



Direct Search in the Solution Space

- Representations of a solution space can be roughly categorized as:
  - Continuous
  - Discrete
    - Binary
    - Integer
    - Permutation
    - …
- Different representations may favor different search methods, but most of them share a common framework.

General Search framework



General Search Methods

- **Typical Frameworks:**
  - Local Search
  - Simulated Annealing
  - Tabu Search
  - Population-based search

- **Two basic issues (differs over concrete search methods):**
  - search operator (how to generate a new candidate solution)
  - evaluation criterion (or replacement strategy)

Typical Search Operators

- **A Search Operator generate a new solution based on previous ones.**

$$\phi\colon \mathrm{x} \to \mathrm{x}', \forall \mathrm{x}, \mathrm{x}' \in \mathcal{X}$$

- **"Un-informed" Search Operators in the solution space: sampling**
  - For a binary string: Flip 1 (or n) bit
  - For a permutation: Exchange the positions of two elements
  - For a continuous number/vector: sample a Gaussian distribution

- **In the rest of this lecture, we take continuous case as an example.**

Greedy Local Search Framework

- **A greedy-type search framework in the solution space**
  - Given a predefined Local Search Operator
  - Iteratively generate new solutions
  - Always pick the best solution so far, sometimes also known as Hill Climbing.

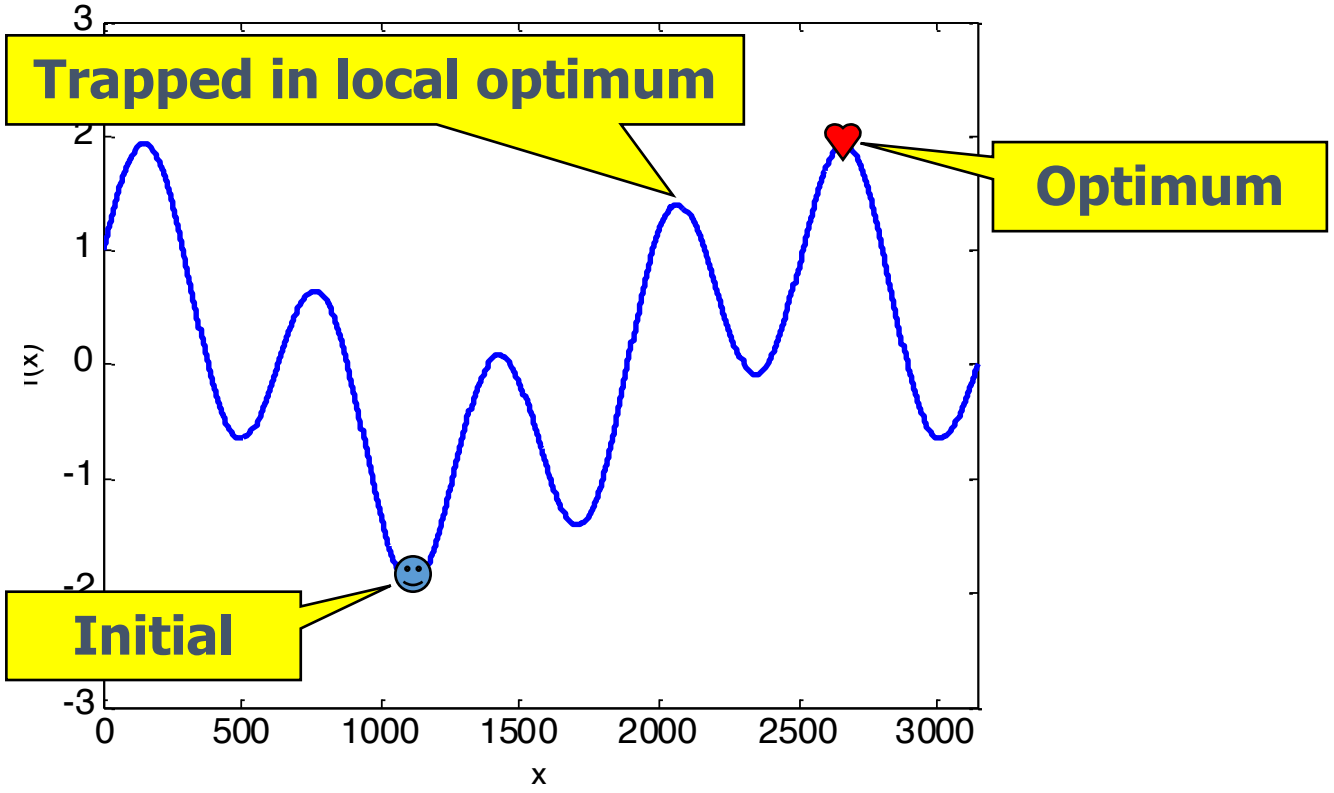
Do while (halt condition is not satisfied)
- Generate solution $\mathbf{x}'$ based on $\mathbf{x}(t)$
- Evaluate the $f(\mathbf{x}')$
- If $f(\mathbf{x}')>f(\mathbf{x}_i)$,
- Then $\mathbf{x}(t+1) = \mathbf{x}'$, otherwise $\mathbf{x}(t+1) = \mathbf{x}$
- $i=i+1$

Greedy~
Hill Climbing~
Steepest Ascent

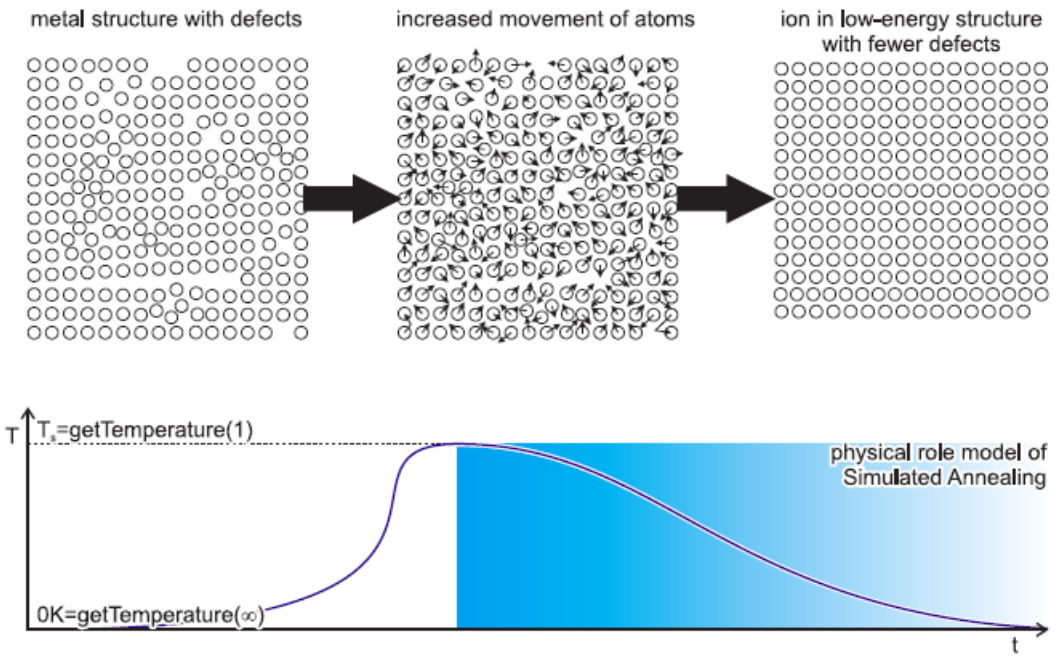
assume a maximization problem

Greedy idea may fail

- On multimodal problems



Simulated Annealing



temperature from high to low
when high temperature, form the shape
when low temperature, polish the detail

Simulated Annealing

- **Hill Climbing → Simulated Annealing**

Replace this line with a probability *p*

Do while (halt condition is not satisfied)

- Generate solution **x**' based on **x**(*t*)
- Evaluate the *f* (**x**')
- If $f(\mathbf{x}') > f(\mathbf{x}_i)$,
- Then **x**(*t*+1) = x', otherwise **x**(*t*+1) = x
- $i = i+1$

$$p = \begin{cases} 1 & if \quad f(x_i) < f(x_i') \\ \exp(-\dfrac{f(x_i) - f(x_i')}{T}) & if \quad f(x_i) \geq f(x_i') \end{cases}$$

Simulated Annealing

- Generate an initial $\mathbf{x}(0)$
- Set the initial temperature $T(0)$;
- Do while (halt condition is not satisfied)

  Get current temperature $T(i)$;

  Generate solution x' based on $\mathbf{x}(i)$

  Calculate f(x')以及$\Delta f = f(\mathbf{x}(i)) - f$(x')

  If $\Delta f < 0$ (for maximization) ， then $\mathbf{x}(i+1)$= x'

  Otherwise, $p = \exp(-\Delta f / T(i))$;

  If c = random[0,1] < p, $\mathbf{x}(i+1)$= x';

  Otherwise $\mathbf{x}(i+1)= \mathbf{x}(i)$;

  $i=i+1$

  Update $T(i)$;
- End Do

To be continued