Monte Carlo Methods (II)

俞 声

清华大学统计学研究中心

Recall the Bayesian inference problem

- The *posterior* distribution of θ given x is

$$f_{\theta|x}(\theta \mid x) = \frac{f_{x|\theta}(x \mid \theta) f_\theta(\theta)}{f_x(x)} = \frac{f_{x|\theta}(x \mid \theta) f_\theta(\theta)}{\int f_{x|\theta}(x \mid \theta) f_\theta(\theta) d\theta}.$$

- $E_{\theta|x}[g(\theta)] = \int g(\theta) f_{\theta|x}(\theta \mid x) d\theta$ can be approximated with Monte Carlo integration – the mean of samples from $f_{\theta|x}(\theta \mid x)$. (the second "MC" in "MCMC")

- However, $f_{\theta|x}(\theta \mid x) = \frac{f_{x|\theta}(x|\theta) f_\theta(\theta)}{\int f_{x|\theta}(x|\theta) f_\theta(\theta) d\theta}$ is hard to compute. How do we generate samples from this distribution?

Recap of Markov chains

- A discrete-time, finite-state Markov chain (for simplicity) is a sequence of random variables $X_1, X_2, X_3, \ldots$ with the *Markov property*, namely that the probability of moving to the next state depends only on the present state and not on the previous states:

$$P(X_{t+1} \mid X_1, X_2, \ldots, X_t) = P(X_{t+1} \mid X_t).$$

- For *time-homogeneous Markov chains* (or *stationary Markov chains*), the transition probability is independent of t. Denote $P(X_{t+1} = j \mid X_t = i)$ with p_{ij}, and $P(X_{t+k} = j \mid X_t = i)$ with $p_{ij}^{(k)}$.

Recap of Markov chains

- A Markov chain is *irreducible* if for all $i, j \in S, i \neq j$, there is $k \geq 1$ such that $p_{ij}^{(k)} > 0$.
- A Markov chain is *aperiodic* if for all $i \in S$, there is $k \geq 0$ such that $p_{ii}^{(k)} > 0$ and $p_{ii}^{(k+1)} > 0$.
- A Markov chain is *positive recurrent* if for all $i \in S$, $\sum_k k \cdot p_{ii}^{(k)} < \infty$.
- A finite-state, aperiodic, irreducible Markov chain has a limiting distribution $\lim_{t\to\infty} P(X_t = j) = \pi_j$, which is also a *stationary distribution*: $\pi_j = \sum_i \pi_i p_{ij}$ for all j.

The Metropolis-Hastings sampler

- The Metropolis-Hastings algorithms allow us to generate a Markov chain (the first "MC" in "MCMC") $\{X_t \mid t = 0, 1, 2 \ldots\}$ such that its stationary distribution is the target distribution f (for our purpose, $f = f_{\theta|x}(\theta \mid x)$).
  - Properties are redefined for the continuous state space. E.g., a stationary distribution is defined as $f(y) = \int_{x \in S} f(x)p(x, y)dx$.

- Two requirements for using the Metropolis-Hastings algorithm:
  - We need a *proposal distribution* (or *jumping distribution*) $g(X \mid X_t)$ for generating the next sample X_{t+1};
  - We need to be able to compute f up to a constant scale.

The proposal distribution

- The choice of proposal distribution g is very flexible, but the chain generated by this choice must have the follow properties:
  - Irreducibility
  - Positive recurrence
  - Aperiodicity

- **Rule of thumb**: A proposal distribution with the same support set as the target distribution will usually satisfy these regularity conditions.

The Metropolis-Hastings sampler

1. Choose a proposal distribution $g(\cdot\,|X_t)$ and a starting point X_0.
2. Repeat until the chain has converged to a stationary distribution according to some criterion:
   a) Generate Y from $g(\cdot\,|X_t)$;
   b) Generate U from Uniform(0,1);
   c) If
   $$U \leq \frac{f(Y)g(X_t \mid Y)}{f(X_t)g(Y \mid X_t)},$$
   accept Y and let $X_{t+1} = Y$; otherwise let $X_{t+1} = X_t$;
   d) Increment t.

Correctness of the M-H algorithm

- A homogenous Markov chain with transition kernel density function $p(x,y) = p(X_{t+1} = y \mid X_t = x)$ is said to satisfy the *detailed balance* or *reversibility* condition if there exists a function f such that:
$$f(x)p(x,y) = f(y)p(y,x)$$
for every pair of x and y.
- This condition can be loosely understood as the process moves from x to y equals to the moves from y to x.
- **Theorem:** Suppose that a Markov chain with the transition function p satisfies the detailed balance condition with f being a pdf. Then, f is a stationary probability density of that chain.
- **Theorem:** The Markov chain produced by the M-H algorithm satisfies the detailed balance condition with f .

Compute the density up to a scale

- Observe that in our case,

$$f = f_{\theta|x}(\theta \mid x) = \frac{f_{x|\theta}(x \mid \theta) f_\theta(\theta)}{f_x(x)} = \frac{f_{x|\theta}(x \mid \theta) f_\theta(\theta)}{\int f_{x|\theta}(x \mid \theta) f_\theta(\theta) d\theta}.$$

- So

$$\frac{f(Y)}{f(X_t)} = \frac{f_{x|\theta}(x \mid Y) f_\theta(Y)}{f_{x|\theta}(x \mid X_t) f_\theta(X_t)}.$$

- The normalizing constant $\int f_{x|\theta}(x \mid \theta) f_\theta(\theta) d\theta$ is canceled out!

- $f_{x|\theta}(x \mid \theta) f_\theta(\theta)$ usually has a simple form and is easy to compute!

Exercise

- Use the Metropolis-Hastings sampler to generate a sample from a Rayleigh distribution: $f(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$, $F(x) = 1 - e^{-\frac{x^2}{2\sigma^2}}$, and $F^{-1}(x) = \sigma\sqrt{-2\log(1-x)}$. For the exercise, let $\sigma = 4$.

- For the proposal distribution, try the Chi-squared distribution with degrees of freedom X_t.

- Evaluate the generated sequence with a quantile-quantile plot.

- Compare the histogram of the generated sequence with the pdf.

The Metropolis sampler

The Metropolis-Hastings sampler is a generalization of the Metropolis sampler. In the Metropolis algorithm, the proposal distribution is symmetric: $g(Y \mid X) = g(X \mid Y)$.

Therefore, in Step 2 c), we only need to compare

$$U \leq \frac{f(Y)g(X_t \mid Y)}{f(X_t)g(Y \mid X_t)} = \frac{f(Y)}{f(X_t)}.$$

The random walk Metropolis sampler

- The random walk Metropolis sampler is a special case of Metropolis samplers.

- Suppose the candidate point Y is generated from a symmetric proposal distribution $g(Y \mid X_t) = g(|X_t - Y|)$. Then at each iteration, a random increment Z is generated from $g(\cdot)$, and Y is defined by $Y = X_t + Z$.

- For example, $Y \mid X_t \sim N(X_t, \sigma^2)$, then $Z \sim N(0, \sigma^2)$.

Intuition of the M-H algorithms

- The Metropolis-Hastings samplers tend to accept points that are increasingly likely under the target distribution.
- Take the Metropolis sampler for example. If $f(Y) \geq f(X_t)$, then the new point Y is always accepted. If $f(Y) < f(X_t)$, then there is a chance $1 - f(Y)/f(X_t)$ that the new point is rejected.
- Therefore, the sampler tend to accept points from the high density region, and occasionally accept points from low density region.
- In the long run, the points will gradually follow the target distribution.

Caveats for the M-H algorithms

- The samples generated by the M-H samplers are correlated. This means that if we want a set of independent samples, we have to throw away the majority of samples and only take every nth sample.

- Autocorrelation can be reduced by increasing the jumping width (related to the variance of the jumping distribution), but this will also increase the likelihood of rejection of the proposed jump.

- In multivariate distributions, finding a jumping size "just right" for all dimensions at once to avoid excessively slow mixing can be very difficult. An alternative approach that often works better in such situations is called *Gibbs sampling*.

The Gibbs sampler

- The Gibbs sampler was invented by Geman brothers, named after Josiah Gibbs.

- Let $X = (X_1, \ldots, X_d)$ be a random vector in $\mathbb{R}^d$. Define the $d-1$ dimensional random vectors

$$X_{(-j)} = \left(X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_d\right)$$

- Denote the corresponding univariate conditional density of X_j given $X_{(-j)}$ by $f(X_j \mid X_{(-j)})$.

The Gibbs sampler

1. Initialize $X(0)$ at time $t = 0$;
2. Staring from $t = 1$; repeat until the chain has converged:
   a) Set $x = X(t-1)$;
   b) For each coordinate $j = 1, \ldots, d$:
      (a) Generate $X_j^*(t)$ from $f(X_j \mid x_{(-j)})$;
      (b) Set $x_j = X_j^*(t)$;
   c) Set $X(t) = \left(X_1^*(t), \ldots, X_d^*(t)\right)$ (no rejection);
   d) Increment t.

在生成X2的时候X1要用新的了

Exercise

- Generate a bivariate normal distribution with mean vector $(\mu_1, \mu_2) = (0,2)$, variances $\sigma_1^2 = 1$, $\sigma_2^2 = 0.25$, and correlation $\rho = -0.75$, using Gibbs sampling.

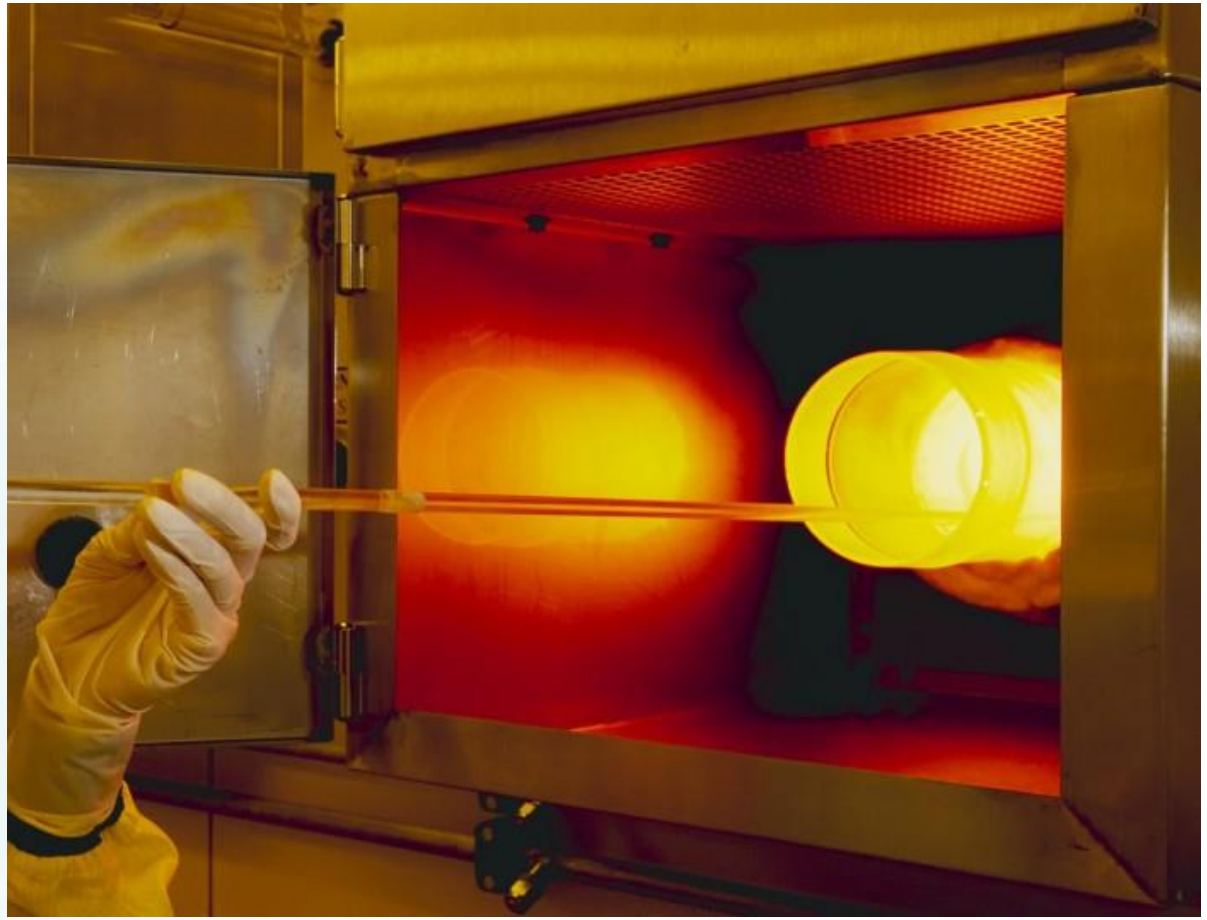
- Know that:

$$f(x_1 \mid x_2) \sim N\left(\mu_1 + \frac{\rho\sigma_1}{\sigma_2}(x_2 - \mu_2), (1-\rho^2)\sigma_1^2\right)$$

$$f(x_2 \mid x_1) \sim N\left(\mu_2 + \frac{\rho\sigma_2}{\sigma_1}(x_1 - \mu_1), (1-\rho^2)\sigma_2^2\right)$$

Simulated Annealing

Heat treatment of metal



Heat treatment of metal

Forging a sword for the movie “Thor”

Heat treatment of metal

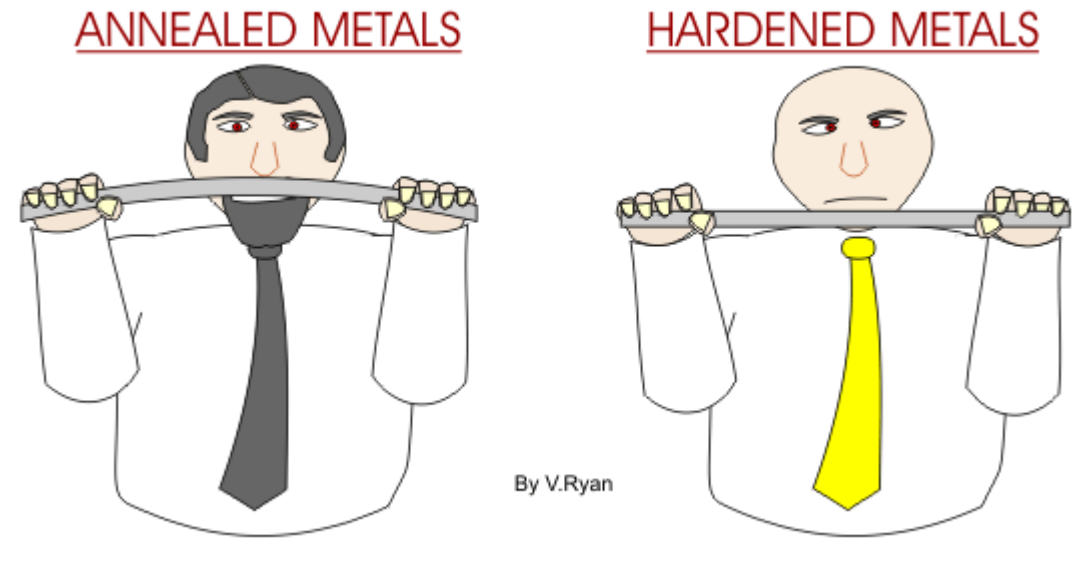
- **Annealing** is frequently used to soften metals including iron, steel, copper, brass and silver. The process involves heating the metal to a specific temperature then allowing it to **cool slowly at a controlled rate**.

- **Normalizing** is applied to alloys to provide uniformity in grain size and composition. The metal is heated to a predefined temperature then **cooled by air**.

- **Hardening** is applied to steel and other alloys to improve their mechanical properties. During hardening, the metal is heated at a high temperature. Next the metal is quenched, which involves **rapidly cooling it in oil or water**.

- **Tempering** is a low temperature heat treatment process normally performed **after hardening** in order to reach a desired hardness/toughness ratio.

Heat treatment of metal



Statistical mechanics behind annealing

The Boltzmann distribution gives the probability that a system will be in a certain state as a function of that state's energy

$$p_i = \frac{e^{-E_i/kT}}{\sum_j e^{-E_j/kT}} \propto e^{-E_i/kT}$$

where E_i is the energy of state i, k the Boltzmann's constant, and T the thermodynamic temperature.

- Dislocation of atoms – high energy – hard & brittle;
- Remove dislocations – low energy – soft & ductile.

Statistical mechanics behind annealing

- The Boltzmann distribution shows that states with lower energy will have a higher probability of being occupied than the states with higher energy.

- The relative probabilities are heavily affected by the temperature.

```
set.seed(125)
E = runif(10,0,10)
f = exp(-E/T), T = 100, 10, 5, 1, 0.5, 0.1
f = f/sum(f)
plot(f, ylim=c(0,1))
```

Statistical mechanics behind annealing

If the distribution says the system will almost surely occupy the state with the lowest energy when the temperature is near zero, why do we need to heat the metal first?

Because the distribution is the stationary distribution of a Markov chain. The actual state of the system depends on where the state was initially.

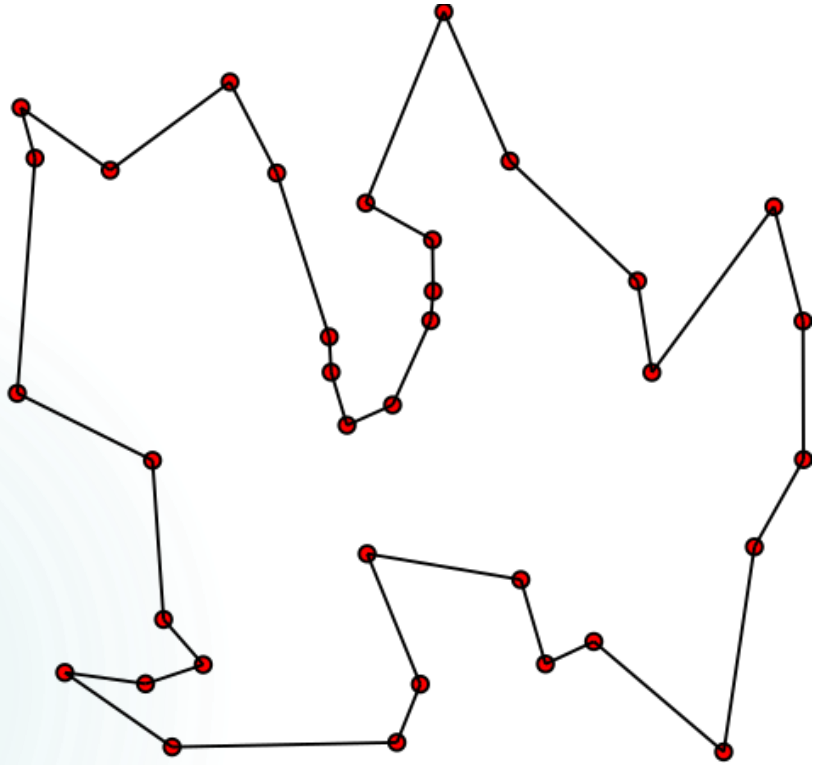
- Heating allows the system to shift to the low energy state.
- Cooling locks the system state. If cooled too quickly, the system may still be at a high-energy state.

Simulated annealing for optimization

- Think $f(x)$ as E_i. Thus, $P_X(X = x) \propto e^{-f(x)/T}$. k is omitted.

- $f(x)$ doesn’t have to be continuous. It can well be a combinatorial optimization, such as the “traveling salesman problem”.

The traveling salesman problem



Question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city? (NP-hard)

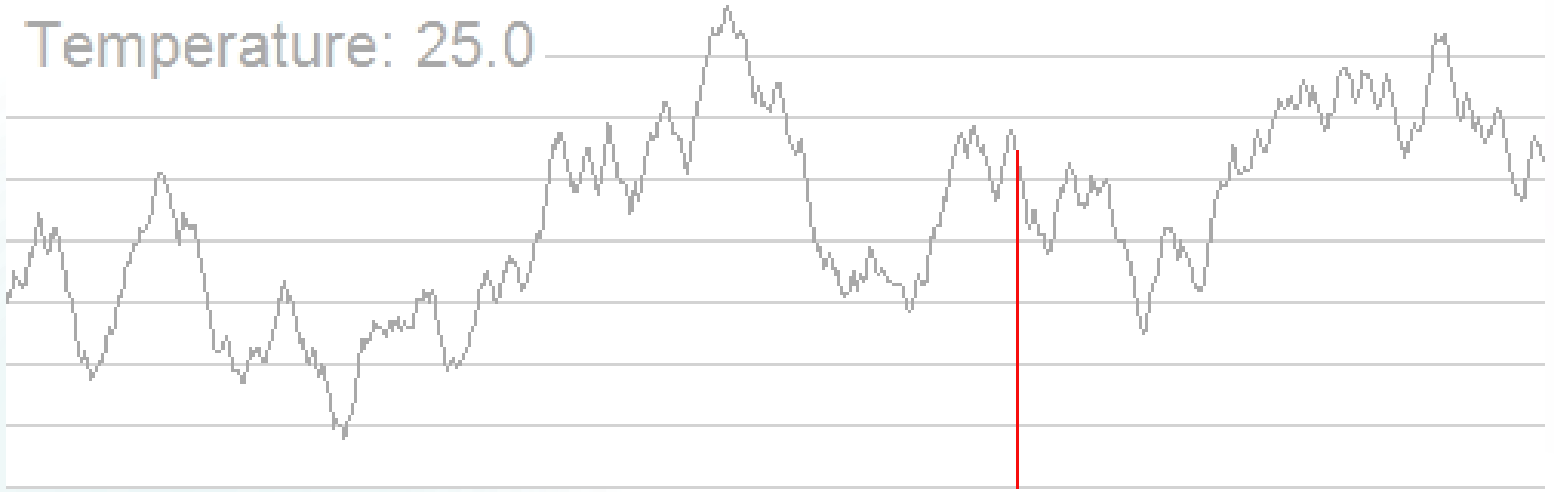
Simulated annealing for optimization

- Think $f(x)$ as E_i. Thus, $P_X(X = x) \propto e^{-f(x)/T}$. k is omitted.
- $f(x)$ doesn't have to be continuous. It can well be a combinatorial optimization, such as the "traveling salesman problem".
- We can generate a Markov chain with stationary distribution P_X with Metropolis sampling.
  - We need a proposal distribution to get X_{i+1} from X_i.
  - Acceptance rate is $\min\left\{1, \exp\left(\frac{f(X_i)-f(X_{i+1})}{T}\right)\right\}$.
- Slowly reduce T to lock X with a high probability at the minimum point.

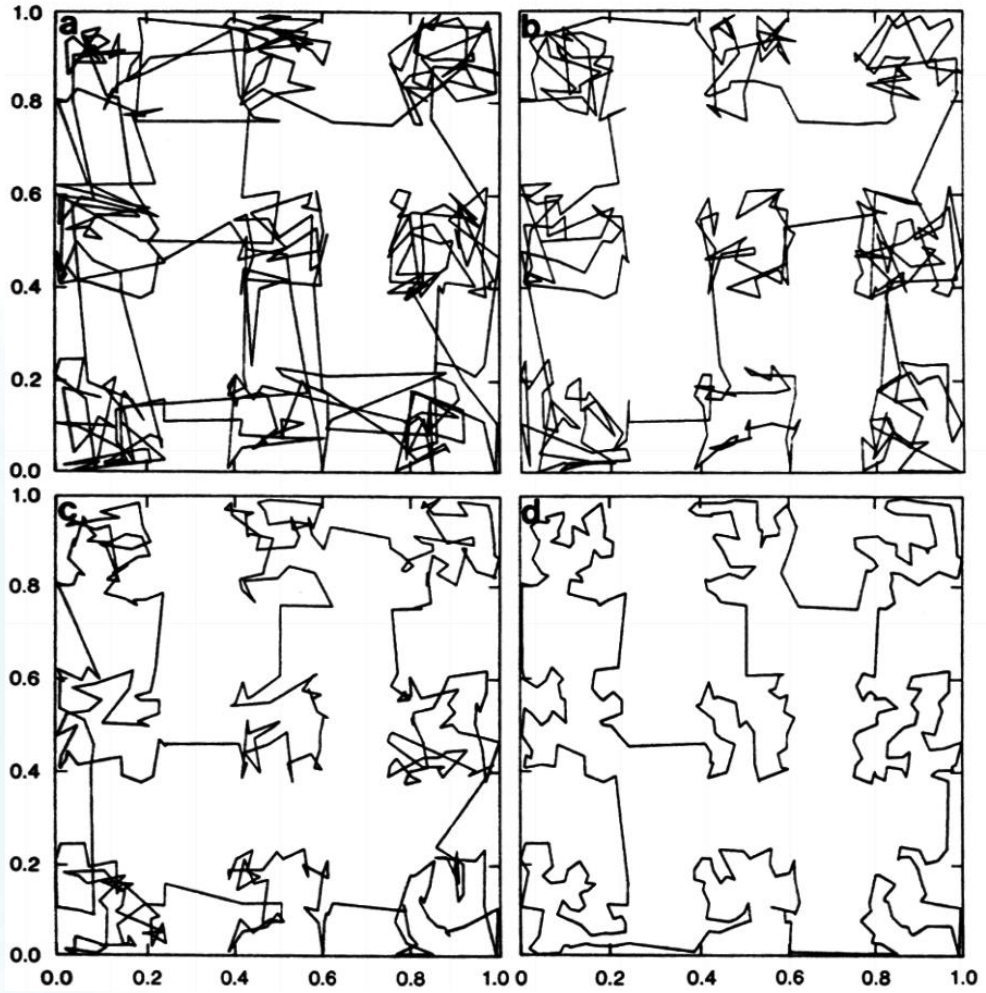
Pseudo-code

```
Create random initial solution γ
E_old = cost(γ);
for(temp=temp_max; temp>=temp_min; temp=next_temp(temp)) {
    for(i=0;i<i_max; i++ ) {
      γ_new = succesor_func(γ); //a randomized function
      E_new = cost(γ_new);
      if(random() < exp(-(E_new-E_old)/temp)) { //accept
          E_old = E_new;
          γ = γ_new;
      }
    }
}
```

The traveling salesman problem



The traveling salesman problem



Simulated annealing solving a 400-city TSP

Practical issues

- SA requires a huge number of function evaluations.
  - Thus not a good choice when you can use linear programming, convex programming, etc.
- Designing the cooling schedule is an *art*.
  - Too fast – suboptimal solution;
  - Too slow – too many function evaluations.
- Designing the proposal distribution is also an *art*.
  - Problem inherited from Metropolis sampling.