Lecture 13: Simulated Annealing

1. Review of the last lecture:
   - Penalty functions and function transformation.
   - Stochastic ranking for constraint handling
   - Repairing methods, Lamarckian evolution and Baldwin effect
2. What is simulated annealing
3. Simulated annealing and neighbourhood search
4. Dynamic neighbourhood size
5. Summary

Physical Annealing and Metropolis Procedure

1. The idea comes from condensed matter physics.
2. The annealing technique in physics: heat a solid first to a high temperature, then cool it down slowly (especially slow around the freezing point of the solid). We should get a solid in its ground state (crystal), i.e., a state with the minimum energy.
3. Metropolis procedure (1953): A procedure to simulate the above annealing process. One iteration of the procedure is as follows:
   (a) Generate a new state from the current one by giving a random displacement to a randomly selected particle.
   (b) Accept the new state if the energy of the new state is no greater than that of the current one, otherwise, it will only be accepted with probability
   $$\exp\left(-\frac{E_{new\_state} - E_{current\_state}}{T}\right)$$
   where E stands for the energy of the system and T is the temperature.

Annealing and Optimisation

1. Metropolis procedure illustrates clearly that the annealing process is equivalent to the process of minimising energy E.
2. The analogy between finding minimum energy states in a physical system and finding minimum cost configurations in a combinatorial optimization problem was independently observed by Kirkpatrick et al. and Černy.
3. Simulated annealing can be regarded as the application of the Metropolis procedure to optimisation problems.

Combinatorial Optimisation and Neighbourhood Structures

Combinatorial Optimization: Given a finite configuration space $S = \{X \mid X = (x_1, x_2, \cdots, x_m)\}$, where m is called the dimension of the space, and a cost function $c: S \longrightarrow R^1$, we want to find an optimum configuration $X \in S$, such that $\forall Y \in S, c_X \leq c_Y$.

Neighbourhood Structure: The neighbourhood system of a combinatorial optimization problem can be defined as

$$\mathcal{N} = \{N_X \mid X \in S, N_X \subseteq S\} \tag{1}$$

where $X \notin N_X$ and $Y \in N_X \Longleftrightarrow X \in N_Y$. Obviously, elements in N_X represent the configurations next to X in the configuration space. They are called neighbours of X.

Classical Simulated Annealing

select initial configuration X at random;
select initial temperature T;
REPEAT
  REPEAT
    randomly select Y from N_X with uniform distribution;
    IF $c_Y \leq c_X$
      THEN accept Y as the new configuration
      ELSE accept Y as the new configuration with probability
        $\exp\left(-\left(c_Y - c_X\right) / T\right)$
  UNTIL 'inner-loop stop criterion' satisfied;
  decrease temperature T
UNTIL 'outer-loop stop criterion' satisfied

Does Simulated Annealing Work?

1. SA has been **shown** to converge to the global optimum as the temperature T decreases **sufficiently slow** to 0.
2. **shown**: The proof consists of three steps:
   - *Weak Ergodicity:* There is a positive probability of visiting any state (configuration) in the space.
   - *Strong Ergodicity:* The Markov chain converges to a stationary distribution. (The algorithm will stay at certain states eventually.)
   - *Convergence:* The stationary distribution is the optimal one. (The algorithm will stay at where we are looking for.)
3. **sufficiently slow**: $T_n = \frac{h}{\ln(n+1)}, n = 1, 2, \cdots$, where h is a problem dependent constant and T_n stands for the temperature at time n.

Notations (Not Math)

Suppose the probability of generating the next configuration Y from the current one X is g_{XY}, which is independent of temperature T, and the probability of accepting Y as the new current configuration is $a_{XY}(T)$.

$$g_{XY} = \begin{cases} \frac{1}{|N_X|}, & X \in S, Y \in N_X \\ 0, & X \in S, Y \notin N_X \end{cases} \tag{2}$$

$$a_{XY}(T) = \min\left\{1, \exp\left(-\frac{c_Y - c_X}{T}\right)\right\} \tag{3}$$

More Notations (Still No Math)

The one-step transition probabilities of the nonstationary Markov chain associated with CSA:

$$p_{XY}(T) = \begin{cases} \frac{1}{|N_X|} \min\left\{1, \exp\left(-\frac{c_Y - c_X}{T}\right)\right\}, & X \in S, Y \in N_X \\ 0, & X \in S, Y \notin N_X, X \neq Y \\ 1 - \sum_{Z \in N_X} \frac{1}{|N_X|} \min\left\{1, \exp\left(-\frac{c_Z - c_X}{T}\right)\right\}, & X \in S, Y \notin N_X, X = Y \end{cases}$$

Well, it tells you how likely CSA is going to move from X to Y.

A Mathematical Result Without Doing the Math

Given that CSA's generation and acceptance probabilities satisfy (2) and (3) respectively, then CSA converges to global minima, i.e.,

$$\lim_{n\to\infty} Prob\,\{X \in S^*\} = 1 \tag{4}$$

$$S^* = \{X \mid X \in S, c_X \leq c_Y \forall Y \in S\} \tag{5}$$

if the rate of temperature decrease is

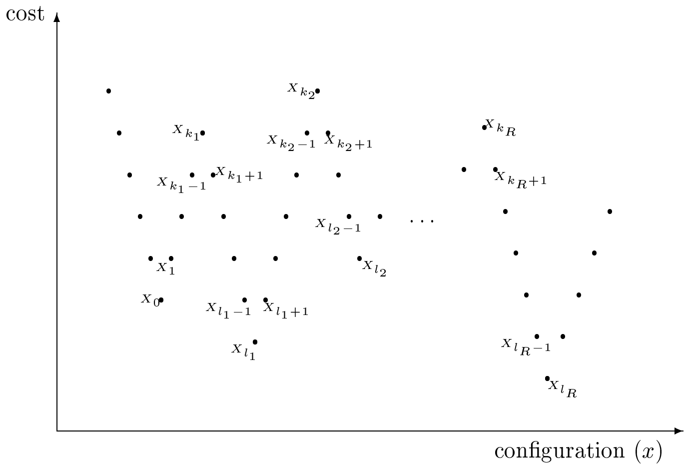
$$T_n = \frac{h}{\ln(n+1)}, \qquad n = 1, 2, \cdots \tag{6}$$

where h is a problem dependent constant and T_n stands for the temperature at time n.

Discussions (No More Math)

1. The theoretical cooling schedule is far too slow to be practical.
2. In practice, we use
$$T_{n+1} = \rho T_n \tag{7}$$
where ρ is a constant normally selected between 0.80 and 0.99.
3. There have been many efforts in increasing the cooling schedule and speeding up CSA.
4. However, there are other things we can do to improve CSA's performance.

A Simple Example of 1-d Problem



Analytical Results

1. In the one-dimensional case, $|N_{X_i}| = 2$ for all $X_i \in S$.
2. We can calculate the probability of finding the global minimum X_{l_R} from configuration X_0 using no more than l_R steps:
$$Prob\,(X_0, X_{l_R}, l_R, T, |N_{X_i}| = 2) = \left(\frac{1}{2}\right)^{l_R} \exp\left(-\frac{1}{T}\sum_{i=1}^{R}\left(c_{X_{k_i}} - c_{X_{l_{i-1}}}\right)\right) \tag{8}$$
where $l_0 = 0$.
3. There are three factors which affect the probability of reaching a global minimum from a non-global minimum: (1) the total height of the hills which have to be surmounted; (2) the temperature; and (3) the total probability of generating the global minimum from the non-global minimum, i.e.,
$$\left(\frac{1}{2}\right)^{l_R} \tag{9}$$

The Impact of Neighbourhood Size

1. Suppose we have extended the neighbourhoods from $N_{X_i} = \{X_{i-1}, X_{i+1}\}$ to $N_{X_i} = \{X_{i-2}, X_{i-1}, X_{i+1}, X_{i+2}\}$.
2. We can get the following result after some calculations:

$$Prob\left(X_0, X_{l_R}, l_R, T, |N_{X_i}| = 4\right) > Prob\left(X_0, X_{l_R}, l_R, T, |N_{X_i}| = 2\right) \quad (10)$$

   where $l_R \geq 2$.
3. In general,

$$Prob\left(X_0, X_l, l, T, |N_{X_i}| = 2m\right) > Prob\left(X_0, X_l, l, T, |N_{X_i}| = 2\right) \quad (11)$$

   where $l \geq m$.

Is it always the case that the larger the better?

Dynamic Neighbourhood Size: Idea

1. General rule of thumb: large neighbourhood sizes in the beginning to explore the whole space, and small neighbourhood sizes later to exploit a small region.
2. In the following discussion, we say that the "Hamming " distance between two configuration $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$ is ξ if there are exactly ξ different elements between them.
3. The idea: Use a continuous density function $f(x)$ to generate the distance between the current and next configurations.

Dynamic Neighbourhood Size: How?

Denote the set of configurations which are ξ distance from the current configuration X as $S_X(\xi)$,

$$S_X(\xi) = \{Y \mid Y \in S, d_{XY} = \xi\} \quad (12)$$

The probability of generating configuration Y, which is d_{XY} distant from configuration X, is defined as

$$g_{XY}(T_n) = \frac{f(d_{XY}) \,/\, |S_X(d_{XY})|}{F_X(T_n)} \quad (13)$$

where

$$F_X(T_n) = \sum_{d_{XZ}=1}^{d_{max}} \sum_{Z \in S} \frac{f(d_{XZ})}{|S_X(d_{XZ})|} \quad (14)$$

Empirial Results

TSPs with 40 cities were used in our experiment and were generated at random.

problem instance	1	2	3	4
initial value	15080	12260	13760	15820
NorSA	2540	2140	2560	2300
CSA	3120	2520	2880	2460

Table 1: Comparison of SA with a fixed neighbourhood size (CSA) and SA with a dynamic neighbourhood size (NorSA). Normal distribution is used to generate the neighbourhood size.

Summary

1. Similarities and differences between SA and evolutionary algorithms.
2. Neighbourhood size is very important in SA.
3. SA can be analysed vigorously.
4. SA can be applied to combinatorial as well as numerical optimisation.
5. SA can be combined with other algorithms, e.g., genetic annealing.
6. Fast SA, very fast SA, ...
7. Applications: VLSI cell placement, routing, facility layout, scheduling (e.g., one-day cricket test match scheduling), frequency assignment for mobile phones, road planning, school timetabling, ...

References

1. X. Yao, "Simulated annealing with extended neighbourhood," *International Journal of Computer Mathematics*, **40**:169-189, 1991.
2. X. Yao, "Comparison of different neighbourhood size in simulated annealing," In *Proc. of the Fourth Australian Conf. on Neural Networks (ACNN'93)*, ed. P. Leong and M. Jabri, Melbourne, pp.216–219, 1993.