TIES451
Selected Topics on Soft Computing
Mou Hao

Exercise 1 - Simulated Annealing

Overview

Simulated annealing is a probabilistic method proposed in Kirkpatrick, Gelett and Vecchi (1985) and Cerny (1985) for finding the global minimum of a cost function that may possess several local minima. It works by emulating the physical process whereby a solid is slowly cooled so that when eventually its structure is “frozen” this happens at a minimum energy configuration.

As hinted by the process of annealing of metal, people produce simulated annealing algorithm to find global minimum. The idea is that according to the temperature, there is some possibility that the position could change randomly, somehow.

Algorithm：

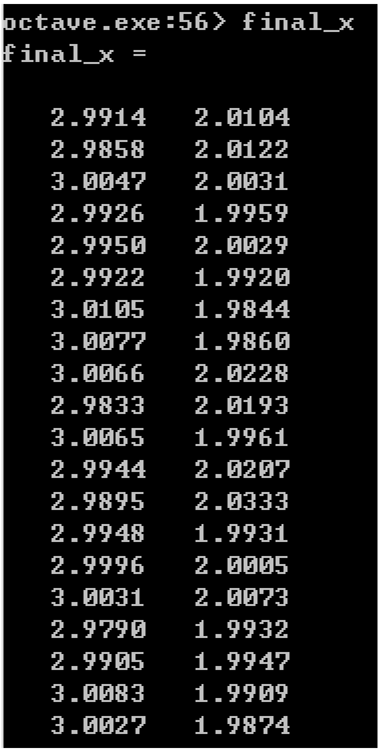
```
0：
    select initial configuration x
    set initial temperature T
1:
    evaluate energy function f1 = f(x)
    generate random change of x -> $\Delta x$    ------------ (1)
    evaluate energy function f2 = f(x+$\Delta$x)
2:
    $\Delta f = f2 - f1$
    if $\Delta f <= 0$
        set $x = x + \Delta x$
    else
        generate random number u between (0,1)
        if $u < e ^ {(-\Delta f / T)}$
            set $x = x + \Delta x$
        else
            $x = x$
3:
    T = f(T), basically we can assume that $T = T - \epsilon$  ---------------------(2)
4:
    if $T > 0$ return to step 1
```

In the above (1) and (2), how to generate random changes of state x, and how to decrease the temperature T, may affect the final result and also the efficiency of this algorithm.

Practice

I am doing an experiment with different initial temperatures, and evaluate the result for this algorithm.

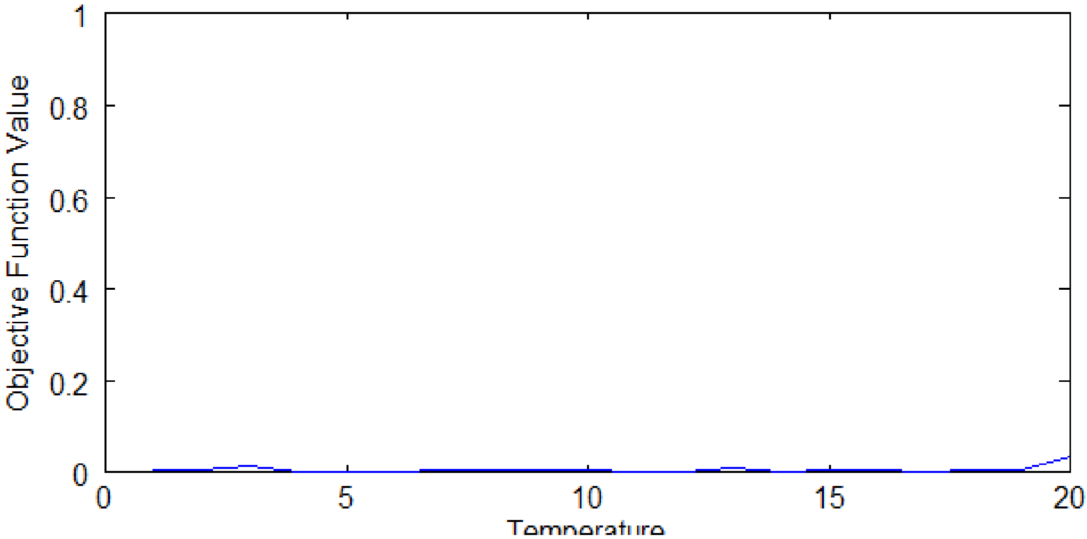
optimum state:



```
octave.exe:56> final_x
final_x =

   2.9914   2.0104
   2.9858   2.0122
   3.0047   2.0031
   2.9926   1.9959
   2.9950   2.0029
   2.9922   1.9920
   3.0105   1.9844
   3.0077   1.9860
   3.0066   2.0228
   2.9833   2.0193
   3.0065   1.9961
   2.9944   2.0207
   2.9895   2.0333
   2.9948   1.9931
   2.9996   2.0005
   3.0031   2.0073
   2.9790   1.9932
   2.9905   1.9947
   3.0083   1.9909
   3.0027   1.9874
```

Plot a figure of final Objective function with different initial temperature. And the results are almost the same.


Objective Function Value
Temperature