268 HW3

Sihong HE #ID:89483859

Problem1

In this problem, I wrote a function simulated_annealing(func, x0, allowed, t0, t_final,jud) which base on the Simulated Annealing Alg.;

Func: the objective function whose variables can be multiple discrete or continues or combinational.

X0: the starting points

Allowed: the range of each components x_i of x

To: the beginning temperature which is defaulted to be 10

T_final: the finally temperature which is defaulted to be 0.1

Jud: the judgement vector for variables when jud_i = 1, components_i is discrect; when jud_i = 0, it's continuous and with specific range; jud_i = 2, it's continuous in R.

And there are some helper functions in the simulated_annealing(). And I choose Nt=5, Nc=200, rt=0.9, rs=0.9, St=1.0.

(a) I test my function with a 0/1 knapsack with a large penalty for overfilling the knapsack problem. The problem is clarified as below:

We have a bag which can hold at most 10 kg items. And we have 6 items their weight and value are ($10.0, 3.3kg), ($1,.001kg), ($1,.005kg), ($1,5.0kg), ($3.0,.01kg), ($5.0,3.0kg). We want to take more valuable items within the capacity of the bag.

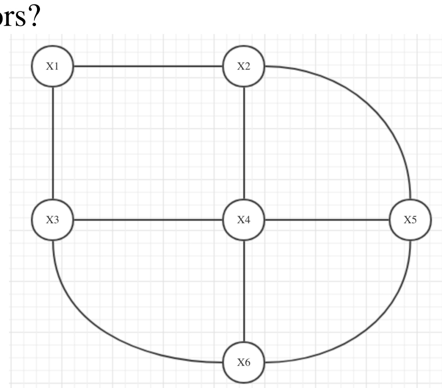
After using my function I found that the optimum of this problem is take all items except item 4. The solution of this problem is [1,1,1,0,1,1]. And we have such table:

	Average error	Function value	# Inner iteration
0/1 knapsack	0 +- 0	-13.34+-1.78e-15	201+-0

After several times testing I found that problem size = 9 is the maximum problem size. When problem size > 9, I can't get the right answer.

(b) I test my function with graph-3-coloring problem. The problem is clarified as below:

Here's a graph. Using only three colors, can we color the vertices such that adjacent vertices have different colors?



It turns out that we can color these nodes using only there colors such that obey problem's condit ion. The solution can be [1, 2, 0, 1, 0, 2].

Problem2

I use the function mentioned in HW3:

$$f(x) = x\sin(x), \text{ in the interval } [-(2N+0.25)\pi, (2N+0.25)\pi]$$

By using my SA function in n = 0, 1, 2, 3, 4 separately, I can get this table:

	Minimum	Function Value	# Inner iteration
N = 0	0	0	201
N = 1	-4.915	-4.814	201
N = 2	-11.115	-11.036	201
N = 3	-17.344	-17.307	201
N = 4	-23.609	-23.583	201

By using my coordinate descent function in my HW1, I can get this table:

N = 0	Minimum	Function Value	# Iteration
Starting = 0	0	0	11
Starting = 4	4.913	-4.814	19
Starting = 8	11.085	-11.041	22
Starting = 12	11.085	-11.041	19
Starting = 16	17.336	-17.308	21
Starting = 20	17.336	-17.308	22
Starting = 24	23.604	-23.583	17

We can find that the SA function can find the global minimum, but the CD function heavily depends on the choice of starting point and usually can only find the local minimum.