CS 268 Intro to Optimization Homework 3

Chen Li

October 27, 2015

1 Problem1

I use the Metropolis approach for Simulated Annealing, and set the initial temperature $T = 50$ and its decay ratio equal to 0.98. When $T < 0.1$ the loop ends.

1.1 1(a)

I choose the graph-2-coloring problem to solve. Firstly I test my optimizer on a simple 5 nodes graph to validate its correctness. Then I use the different size of check board to test the optimizers scalability, Every size I loop 10 times to get the mean of error and error bar. The results are as follows, we can see that the error begins at 4×4 check board, and grow up with the size.

T	$Size$	$error_{estimated}$
50	4	0.0 ± 0.0
	9	0.0 ± 0.0
	16	0.80 ± 0.55
	25	2.80 ± 0.95
	36	6.60 ± 1.19
	49	16.30 ± 1.59
	64	20.50 ± 1.58
	81	34.60 ± 2.13
	100	42.70 ± 2.09

1.2 1(b)

I generalize the graph-2-coloring problem to graph-N-coloring problem by simply change the flip approach from flip between 0 and 1 to flip between 0 and N. And I test it on the simple graph to validate is correctness.

2 Problem2

I use Metropolis approach for the float version. I choose the initial step size to be 0.5, when the accept rate q is great than 0.5 or less than 0.3, adjust the ratio accordingly. I set the const factor of step size adjustment to be 1.2. Also, when q is not in the $0.3 - 0.5$ range, the move would not be accepted (this approach seems increase the performance). Also, every time I randomly choose one dimension to flip(apply step). The result shows that the coordinate descent method is more accurate in this situation but it cost more time. I test it on the function $f(x, y) = (x - 1)^2 + (y - 1)^2$.

Function	$Error$	$Time$
SA	0.049	0.0198
$CoordinateDescent$	$1.241e^{-6}$	0.0211

3 Implementation

In $hw3.py$, the $SA()$ implement the Simulated Annealing Method for 2 or more dimension. $SAReal()$ implement the Simulated Annealing Method for real value continues function. The results can be find in the $test\_results$ folder or print on the screen.