

Sep 01, 14 19:09

csc710sbse: hw1:Theisen

Page 1/1

```

#From Class Discussion 8/26/2014
from __future__ import division
import sys, re, random, math
sys.dont_write_bytecode = True

5 kmax = 500

#From Baseline Study - schaffer_trials.py
emax = 44
10 emin = 0

rand = random.random

#Structure from SA Lecture
15 def say(x):
    sys.stdout.write(str(x)); sys.stdout.flush()

    def Energy(x):
        rawAns = math.fabs((x*x) - (x-2)*(x-2))
        ans = (rawAns - emin) / (emax - emin)
        20         return ans

    def Neighbor(x):
        25         return random.uniform(-10, 10)

#Structure from SA Lecture
def main():
    s = random.uniform(-10, 10) #random start
    e = Energy(s)
    30     sBest = s
    eBest = e
    k = 1
    say(int(math.fabs(eBest-1)*100))
    say(' ')
    35     while k < kmax:
        sNew = Neighbor(s)
        eNew = Energy(sNew)
        if eNew < eBest:
            sBest = sNew
            40             eBest = eNew
            say('!!')

        myRand = random.random()

        45         if eNew < e:
            s = sNew
            e = eNew
            say('++')

        #Probability Check from SA Lecture
        50         elif math.exp(-1*(e*eNew)/(k/kmax)) < myRand:
            #P function should be between 0 and 1
            #more random hops early, then decreasing as time goes on
            s = sNew
            e = eNew
            55             say('??')
            #print 'Random Hop! (?)'

        say('.,')
        k = k + 1
        if k % 50 == 0 ^ k != kmax:
            60             print ''
            say(int(math.fabs(eBest-1)*100))
            say(' ')

    print '\nFound best - s:', sBest, ' e:', eBest

65 main()

```

Sep 01, 14 12:53

csc710sbse: hw1:Theisen

Page 1/1

```

#From Class Discussion 8/26/2014
from __future__ import division
import sys, re, random, math
sys.dont_write_bytecode = True

5   kmax = 100000 #100000 run trial for max

    rand = random.random

10  low = 100

    def say(x):
        sys.stdout.write(str(x)); sys.stdout.flush()

15  def Energy(x):
        ans = math.fabs((x*x) - (x-2)*(x-2))
        return ans

    def Neighbor(x):
20      return random.uniform(-10, 10)

    def eMax():
        s = random.uniform(-10, 10) #random start
        e = Energy(s)
        sBest = s
25      eBest = e
        k = 1
        while k < kmax:
            sNew = Neighbor(s)
            eNew = Energy(sNew)
30            if eNew > eBest: #find largest difference
                sBest = sNew
                eBest = eNew
            k = k + 1

35      print 'Found eMax - s:', sBest, ' e:', eBest

    def eMin():
        s = random.uniform(-10, 10) #random start
        e = Energy(s)
        sBest = s
40      eBest = e
        k = 1
        while k < kmax:
            sNew = Neighbor(s)
            eNew = Energy(sNew)
            if eNew < eBest: #find smallest difference
                sBest = sNew
                eBest = eNew
45            k = k + 1

50      print 'Found eMin - s:', sBest, ' e:', eBest

    eMax()
55  eMin()

```