```
from __future__ import division
from searcher import *
from models import *
import sys
from time import gmtime, strftime
import sys, random, math, datetime, time,re
sys.dont_write_bytecode = True

def xtile(lst,lo=0,hi=0.001,width=50,
              chops=[0.1 ,0.3,0.5,0.7,0.9],
              marks=["-" ," "," "," "],
              bar="|",star="*",show=" %3.0f"):
  """The function _xtile_ takes a list of (possibly)
 unsorted numbers and presents them as a horizontal
 xtile chart (in ascii format). The default is a
 contracted _quintile_ that shows the
 10,30,50,70,90 breaks in the data (but this can be
 changed- see the optional flags of the function).
 """
  def pos(p)   : return ordered[int(len(lst)*p)]
  def place(x) :
    return int(width*float((x - lo))/(hi - lo))
  def pretty(lst) :
    return ','.join([show % x for x in lst])
  ordered = sorted(lst)
  lo      = min(lo,ordered[0])
  hi      = max(hi,ordered[-1])
  what    = [pos(p)   for p in chops]
  where   = [place(n) for n in  what]
  out     = [" "] * width
  for one,two in pairs(where):
    for i in range(one,two):
      out[i] = marks[0]
    marks = marks[1:]
  out[int(width/2)]    = bar
  out[place(pos(0.5))] = star
  return ''.join(out) +  "," +  pretty(what)


for x in [Schaffer, Kursawe, Fonseca, ZDT1]:
  eb=50*[None]
  for y in [sa, MaxWalkSat]:
    for r in xrange(1):
      print '\n', x.__name__, y.__name__, '\n', strftime("%a, %d %b %Y %H:%M:%S ",
gmtime())
      a=y(x,disp=False)
      k=x()
      hi, lo, kooling, indepSize, iterations = k.getInit()
      print 'min=', lo, ', max=', hi, ', Cooling Factor=', kooling
      eb[r] =  a.runSearcher()

    #print eb/20
  for x in xrange(50): sys.stdout.write('-')
  sys.stdout.write('\n')
  #xtile(eb)
```

```
# -*- coding: utf-8 -*-
"""
Created on Mon Sep 15 03:04:43 2014

@author: rkrsn
"""
from __future__ import division
import sys
import math, random, numpy as np, scipy as sp
sys.dont_write_bytecode = False
from models import *

# Define some aliases.
rand=random.uniform
randi=random.randint
exp=math.exp

class sa(object):
  def __init__(self,modelName, disp=False):
    self.modelName=modelName
    self.disp=disp
  def runSearcher(self):
    modelbasics=modelBasics(self.modelName);
    modelFunction=self.modelName()
    hi, lo, kooling, indepSize, iterations=  modelFunction.getInit()
    emax, emin = modelbasics.baselining(self.modelName)
    sb=s=[randi(lo,hi) for z in xrange(indepSize)];
    eb=e= modelbasics.energy(s,emax,emin)
    for k in xrange(1,iterations):
      sn=modelbasics.neighbour(s,hi,lo)
      en=modelbasics.energy(sn,emax,emin)
      t=k/iterations
      if en<eb:
        eb, sb=en, sn;
        if self.disp:
          modelbasics.say('!')

      if en<e:
        s, e = sn, en;
        if self.disp:
          modelbasics.say('+')

      elif modelbasics.do_a_randJump(en,e,t,kooling): # The cooling factor needs
 to be reallylow for some reason!!
        s, e=sn, en;
        if self.disp:
          modelbasics.say('?')
      if self.disp:
        modelbasics.say('.')
      if k%40==0:
        if self.disp:
          modelbasics.say('\n')# sa.say(format(sb,'0.2f'))

    if self.disp:
      modelbasics.say('\n'),#modelbasics.say('Best Value Found '), modelbasics.s
ay(sb)

  # Print Energy and best value.
    if self.disp:
      modelbasics.say('\n')
    return eb

class MaxWalkSat(object):
  def __init__(self, modelName, disp=False, maxTries=100, maxChanges=100):
    self.modelName=modelName
    self.disp=disp
    self.maxTries=maxTries
    self.maxChanges=maxChanges
  def runSearcher(self):
    modelbasics=modelBasics(self.modelName);
    modelFunction=self.modelName()
    hi, lo, kooling, indepSize, iterations=  modelFunction.getInit()
    emax, emin = modelbasics.baselining(self.modelName)
```



```
    for i in xrange(self.maxTries):
        # Lets create a random assignment, I'll use list comprehesions here.
        x=xn=xb=[rand(-lo,hi) for z in xrange(indepSize)]
        # Create a threshold for energy, let's say thresh=0.1% of emax (which is
 1) for starters
        thresh=1e-7
        for j in xrange(self.maxChanges):
            # Let's check if energy has gone below the threshold.
            # If so, look no further.
            if modelbasics.energy(xn,emax,emin)<thresh:
                if self.disp:
                  modelbasics.say('.')
                break
            else:
                randIndx=randi(0,indepSize-1) # Choose a random part of solution
 x
                if rand(0,1)<1/indepSize: # Probablity p=0.33
                    y=xn[randIndx]
                    xn[randIndx]=modelbasics.simpleneighbour(y,hi,lo)
                    if self.disp:
                      modelbasics.say('+')
                    #print 'Random change on', randIndx
                else:
                    # xTmp is a temporary variable
                    xTmp= xn; xTmp[randIndx]=rand(lo,hi)
                    xBest=modelbasics.energy(xTmp,emax,emin);
                    # Step from xmin to xmax, take 10 steps
                    Step=np.linspace(lo,hi,10)
                    if self.disp:
                      modelbasics.say('!')
                    for i in xrange(np.size(Step)):
                        xNew=xn; xNew[randIndx]=Step[i];
                        if modelbasics.energy(xNew,emax,emin)<xBest:
                            xBest=modelbasics.energy(xNew,emax,emin)
                            xn=xNew

        if modelbasics.energy(xn,emax,emin)<modelbasics.energy(xb,emax,emin):
          xb=xn
    return modelbasics.energy(xb,hi,lo)

if __name__=='main':
  sa(Schaffer)
```

```
"""
A models file that can be imported to run optimizers
"""
from __future__ import division
import sys
import math, random, numpy as np, scipy as sp
sys.dont_write_bytecode = False
# Define some aliases.
rand=random.uniform
randi=random.randint
exp=math.e
sin=math.sin
sqrt=math.sqrt

class modelBasics(object):
  def __init__(i,model):
    i.model=model()
    i.name=model.__name__
  def do_a_randJump(i, e, en, t, k):
    p=exp**(-(e-en)/(t**k))<rand(0,1)
    return p
  def simpleneighbour(self,x,xmax,xmin):
      return xmin+(xmax-xmin)*rand(0,1)
  def neighbour(i,x,xmax,xmin):
    def __new(x,z):
        return xmin+(xmax-xmin)*rand(0,1) if rand(0,1)<1/(i.model.indepSize) else x[z]
    x_new=[__new(x,z) for z in xrange(i.model.indepSize)]
    return x_new
  def energy(i,x,emax,emin):
    ener=i.model.score(x);
    e_norm= (ener-emin)/(emax-emin)
    return e_norm
  def baselining(i,model):
    emax=0;emin=1;
    indepSize=i.model.indepSize;
    for x in xrange(1000):
      x_tmp=[rand(i.model.baselo,i.model.basehi) for z in xrange(indepSize)]
      ener=i.model.score(x_tmp);
      if ener>emax:
        emax=ener
      elif ener<emin:
        emin=ener
    return emax,emin
  f=open('log_sa_schaffer.txt','w')
  def say(i,x):
    sys.stdout.write(str(x));
    sys.stdout.flush()

class Schaffer(object):

  def __init__(i,hi=100,lo=-100, basehi=1000, baselo=-1000, kooling=0.7, indepSize=1, iterations=2000):
    i.hi, i.lo, i.basehi, i.baselo, i.kooling, i.indepSize, i.iterations= hi, lo, basehi, baselo, kooling, indepSize, iterations
    random.seed()
  def f1(i,x):
    return x*x
  def f2(i,x):
    return (x-2)**2
  def score(i,x):
    return i.f1(x[0])+i.f2(x[0])
  def getInit(i):
    return i.hi, i.lo, i.kooling, i.indepSize, i.iterations

class Kursawe(object):
  def __init__(i,hi=5,lo=-5,kooling=0.6, a=0.8, b=3, indepSize=3, basehi=1000, baselo=-1000, iterations=2000):
    i.hi, i.lo, i.basehi, i.baselo, i.kooling, i.a, i.b, i.indepSize, i.iterations= hi, lo, basehi, baselo, kooling, a, b, indepSize, iterations
    random.seed()
  def f1(i,x):
    return np.sum([-10*exp**(-0.2*sqrt(x[z]**2+x[z+1]**2)) for z in xrange(i.indepSize-1)])
  def f2(i,x):
    return np.sum([abs(x[z])**i.a+5*sin(x[z]**i.b) for z in xrange(i.indepSize)])
  def score(i,x):
    return i.f1(x)+i.f2(x)
  def getInit(i):
    return i.hi, i.lo, i.kooling, i.indepSize, i.iterations

class Fonseca(object):
  def __init__(i,hi=4,lo=-4, basehi=4, baselo=-4, kooling=1.99, indepSize=3, iterations=2000):
    i.hi, i.lo, i.basehi, i.baselo, i.kooling, i.indepSize, i.iterations= hi, lo, basehi, baselo, kooling, indepSize, iterations
    random.seed()
  def f1(i,x):
    return (1-exp**np.sum([(x[z]-1/(np.sqrt(z+1))) for z in xrange(i.indepSize)]))
  def f2(i,x):
    return (1-exp**np.sum([(x[z]+1/(np.sqrt(z+1))) for z in xrange(i.indepSize)]))
  def score(i,x):
    return i.f1(x)-i.f2(x)
  def getInit(i):
    return i.hi, i.lo, i.kooling, i.indepSize, i.iterations

class ZDT1(object):
  def __init__(i,hi=1,lo=0, basehi=1, baselo=0, kooling=1.99, indepSize=30, iterations=2000):
    i.hi, i.lo, i.basehi, i.baselo, i.kooling, i.indepSize, i.iterations= hi, lo, basehi, baselo, kooling, indepSize, iterations
    random.seed()

  def f1(i,x):
    return x[0]
  def g(i,x):
    return (1+9*(np.sum(x[1:]))/(i.indepSize-1))
  def f2(i,x):
    return i.g(x)*(1-sqrt(x[0]/i.g(x)))
  def score(i,x):
    return i.f1(x)+i.f2(x)
  def getInit(i):
    return i.hi, i.lo, i.kooling, i.indepSize, i.iterations
```

```
Schaffer sa
Tue, 16 Sep 2014 11:06:31
min= -100 , max= 100 , Cooling Factor= 0.7
2.50834194422e-08

Schaffer MaxWalkSat
Tue, 16 Sep 2014 11:06:31
min= -100 , max= 100 , Cooling Factor= 0.7
0.866744100907
--------------------------------------------------

Kursawe sa
Tue, 16 Sep 2014 11:06:31
min= -5 , max= 5 , Cooling Factor= 0.6
-0.00184584684969

Kursawe MaxWalkSat
Tue, 16 Sep 2014 11:06:31
min= -5 , max= 5 , Cooling Factor= 0.6
0.00699792004357
--------------------------------------------------

Fonseca sa
Tue, 16 Sep 2014 11:06:31
min= -4 , max= 4 , Cooling Factor= 1.99
-8.14841853689e-11

Fonseca MaxWalkSat
Tue, 16 Sep 2014 11:06:31
min= -4 , max= 4 , Cooling Factor= 1.99
82.9299810715
--------------------------------------------------

ZDT1 sa
Tue, 16 Sep 2014 11:06:34
min= 0 , max= 1 , Cooling Factor= 1.99
0.00597888278735

ZDT1 MaxWalkSat
Tue, 16 Sep 2014 11:06:34
min= 0 , max= 1 , Cooling Factor= 1.99
0.391440210642
--------------------------------------------------
```

```
Schaffer sa
Tue, 16 Sep 2014 11:08:49
min= −100 , max= 100 , Cooling Factor= 0.7
.?.+...........!+.....?.+......!+......?.+.+.+.+....+.
........................................
..............!+...?.+................+......
...+..+............+.......................
........................................
......................................+..
.........?..+..+.....+...+.....+..............
.........+...............................
.....................................?.+.+.
+.+............!+..................?.+.+.+...!+...
........................................
........................................
........................................
......................!+..................
........................................
........................................
........................................
........................................
........................................
........................................
........................................
......................................?.+.
.+.......................................
........................................
........................................
.......+.................................
.......................................+.
...................................+.....
........................................
........................................
....+............?.+......+.................
....+....................................
.............+............?.+.+.+...+.+....+....
..........+..............................
?.+............+.................+..........
.........................?...+............
........................................
...............................+.....?..+..
+.......+........+.........................
..+......................................
........................................
...................................+.....
........................................
........................................
........................................
..........+..............................
.......................?.+.+...............
........................................
........................................
.........?.+.+.+...+.+.......................

2.51382669439e−08
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Kursawe sa
Tue, 16 Sep 2014 11:08:49
min= −5 , max= 5 , Cooling Factor= 0.6
.!+...!+.!+.!+......?..+.+.+..!+.............?.+........
.+..+....+.....!+.........!+........?....!+....?...
.........+........+..!+..?..?...+...+.+......?....
.+.......+...+...................+..........
...!+................!+....!+.................
........................................
......................?..+.!+...............
..?.+...+...+..........+.+....+................
........................................
.................................!+..!+.....
........?...!+...............?..+......+.?..+.+..
+.......+.+..+.....+.........................
.+.................+......+..........!+......
```



```
..................?......................
....+....+................................
...........?...........?..+.+.....+..........
.....+...................?....+......+.+.....
......+.......+...+.........+...............
................?.+.+.+...+..+.....+.......+...?.
...+.........+.+............?.+...........+...
........................+...+.............
........+.........+.......................
........................+........+...?...+..
+...+..................+..........+.........
.........................+...............
........................................
...?......+..........+.....+................
...........+..........+...................
.............+...........................
...........?............................+.
...........+.....................?.+.+......
+.........+...............................
.......................?.................
...........+........................+.....
..............?.+.+..+..+..+.....+.............
........................................
.+..+......+.......................+.....?...
..+............+..+...+.....................
.......+.................................
.....................................?.+..
+..+.......+..+......+.........+....+..........
...............+.........................
..............+..+...+.....+................
........................................
...................+.....................
....?.+....?.................+.....+........+.
.......+..........................+.......
........................+.....+...........
.............................?.+..+.+..+.....
......+....+....+........+......+......+....+.

−0.00191506525987
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Fonseca sa
Tue, 16 Sep 2014 11:08:50
min= −4 , max= 4 , Cooling Factor= 1.99
?.?.!+.?.?.?...+.+.!+.......!+...?..?.+.+.?..+.?.+.+..?.+..+...!+..
....?..+.+...+............!+....!+.........?...+..
.....+.+..........!+..!+.........!+.............
.......!+...!+.......?.+.+...+.+..+.+.+.............
.?....?.?.+..+.+...+........+.+................?..
?.+.+.+.+.....+.+.............................!+.
..!+....................!+................!+..
........................................
.................................!+.......
............!+............................
............................!+............
.................................!+.......
...........!+...........................!+..
........................................
...................?..+.+.+.....+....+........
.......+....+......+...................+..+..
........+...............+.................
..................................+......
.................................+.......
..+......................+................
........................................
.....................+...................
...................+.....................
........................................
...+.....................................
.................................+.......
........................................
........................................
........................................
```