

Oct 06, 14 17:44

csc710sbse:hw4:VivekNair:vnair2

Page 1/2

```

from __future__ import division
import sys
import random
import math
5 import numpy as np
from models import *
from searchers import *
from options import *
from utilities import *
10 from sk import *
sys.dont_write_bytecode = True
#Dr.M
rand= random.random # generate nums 0..1
any= random.choice # pull any from list
15 sqrt= math.sqrt #square root function

def display(modelName,searcher,runTimes,scores,historyhi=[],historylo=[]):
    assert(len(runTimes) == len(scores)), 'Ouch! it hurts'
    print "=====
20 print "Model Name: %s"%modelName
    print "Searcher Name: %s"%searcher
    print "Options Used: ",
    print myoptions[searcher]
    import time
25 print ("Date: %s"%time.strftime("%d/%m/%Y"))
    print "Average running time: %f" % np.mean(runTimes)
    if(len(historyhi)≠0):
        for x in xrange(myModelobjf[modelName]):
            print "Objective No. %d: High: %f Low: %f"%(x+1,historyhi[x],historylo[x])
30 #for i in range(0,len(runTimes)):
    # print "RunNo: %s RunTime: %s Score: %s"%(i+1,runTimes[i],scores[i])
    #print scores
    print xtile(scores,width=25,show=" %1.6f")
    print "=====
35

def multipleRun():
    from collections import defaultdict
    r = 30
    for count in xrange(5):
        myoptions['MaxWalkSat']['probLocalSearch']=(count+1)*0.1
        myoptions['SA']['emax']=(count+1)*0.01
        for klass in [ZDT1:#,Fonseca, Kursawe, ZDT1,ZDT3,Viennet]:
45 eraCollector=defaultdict(list)
        for searcher in [SA,MaxWalkSat]:
            #print "Model Name: %s"%klass.__name__
            #print "Searcher Name: %s"%searcher.__name__
            n = 0.0
            listTimeTaken = []
            listScores = []
            random.seed(6)
            historyhi=[-9e10 for count in xrange(myModelobjf[klass.__name__])]
            historylo=[9e10 for count in xrange(myModelobjf[klass.__name__])]
55 for _ in range(r):
                test = searcher(klass(),"display2")
                print ".",
                import time
                t1 = time.time()
                solution,score,model = test.evaluate()
60 for x in xrange(model.objf):
                    #print len(model.past[x].listing)
                    historyhi[x]=max(model.past[x].historyhi,historyhi[x])
                    historylo[x]=min(model.past[x].historylo,historylo[x])
                    sys.stdout.flush()
65 timeTaken = (time.time() - t1) * 1000
                    listTimeTaken.append(timeTaken)
                    listScores.append(score)
                    eraCollector[searcher.__name__]=listScores
70 #print "Score: %f"%(score)
            print
            callrdivdemo(eraCollector)

```

Oct 06, 14 17:44

csc710sbse:hw4:VivekNair:vnair2

Page 2/2

```

def step2():
75     rdivDemo([
        ["Romantic",385,214,371,627,579],
        ["Action",480,566,365,432,503],
        ["Fantasy",324,604,326,227,268],
        ["Mythology",377,288,560,368,320]])

80

def callrdivdemo(eraCollector):
    print eraCollector
    #print "callrdivdemo %d"%len(eraCollector.keys())
85 keylist = eraCollector.keys()
    print keylist
    variant = len(keylist)
    print variant
    rdivarray=[]
90 for y in xrange(variant):
        #print "Length of array: %f"%len(eraCollector[keylist[y]] [x])
        temp = eraCollector[keylist[y]]
        print temp
        temp.insert(0,str(keylist[y]))
95 print temp
        rdivarray.append(temp)
    rdivDemo(rdivarray)

100 if __name__ == '__main__':
    # random.seed(1)
    # nums = [random.random()*2 for _ in range(100)]
    # print xtile(nums,lo=0,hi=1.0,width=25,show=" %3.2f")
105 # model = ZDT1()
    # model.testgx()
    # for klass in [ZDT1]:
    #     print klass.__name__
    multipleRun()
110 #part6()
    #step2()

```