```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 31 12:42:50 2019

@author: hernan
LIA - MODO International Lab
EA Seminar
Version 1.0 31-07-2019

Example of how to call nsgaii to solve multi-objective knapsack problems
"""


import os
import problem
import nsgaii
import ea_base as ea
import stat_moea
import timeit

def main(ngen, psize,  pc, nvm, clones, function, vhigh):
    """ Main function to call nsgaii to solve multi objective
    0/1 knapasack prpoblems

    Parameters
    ----------
        ngen:      integer       number of generations
        psize:     integer       population size
        pc:        real          crossover rate
        nvm:       integer       number of variables to mutate
        clones:    boolean       true/false
        function:  function name  fitness function to optimize

    Returns
    -------
    Population
        The evolved population

    Comment
    -------
    Some problems
        KP_p-2_n-100_ins-1.dat
        KP_p-3_n-100_ins-1.dat
        KP_p-4_n-40_ins-1.dat
        KP_p-5_n-20_ins-1.dat
    """

    """ Parse command line """


    """ Create the problem """
    mkp = problem.MKP("../problems/" + function + ".dat" )
    fitness=mkp.fitness
    #nvar:      integer        number of varaibles = ntimes
```

```python
        nvar=mkp.nitems
        nobj=mkp.nobj

        s = str(function) + "_nvar"+str(nvar) + "_ngen"+str(ngen)
        s = s + "_psize"+str(psize) + "_pc" + str(pc) + "_nvm" + str(nvm)
        s = s + "_clones" + str(clones) + "_vhigh"+str(vhigh)
        print(s)

        """ Set the output folder and move there"""
        results="../output/Nsgaii/" + s
        os.mkdir(results)
        os.chdir(results)

        """ Run the algorithm a given number of runs """
        ftime = open("time.txt", "w", 1)
        nruns=10
        for i in range(1,nruns+1):
            print("*** Run ", i, " ***")
            run = "run"+str(i)
            os.mkdir(run)
            os.chdir(run)

            tic=timeit.default_timer()
            pop = nsgaii.nsgaii(evaluate = fitness,
                select = ea.binary_tournament_dom_cd, recombine = ea.crossover_1p,
                mutate = ea.bit_flip_mutation, initype='binary', seed=i,
                psize=psize, nobj=nobj, nvar=nvar, vlow=0, vhigh=vhigh, ngen=ngen,
                pcx=pc, pmut=nvm/nvar, keepclones = clones)
            toc=timeit.default_timer()
            ftime.write("Nsgaii run" +str(i)+ " " + str(toc - tic) + " seconds\n")
            os.chdir("..")

        ftime.close()
        """
        Output some statistics in the designated output folder
        """
        nbp= 10 # number of boxplots in addition to gen=0
        gen_list = [int(x*ngen/nbp) for x in range(0,nbp+1)]
        #gen_list=[50, 30, 10,  5, 0]
        """ Reference point for hypervolume """
        refPoint=[0.0]*nobj
        stat_moea.stat_moea(foutput=".", nruns=nruns,
            gen_list=gen_list, nvar=mkp.nitems, nobj=nobj, popsize = psize,
            ngen=ngen, refPoint=refPoint, maxhv=True)
        return pop

if __name__ == "__main__":
    main(ngen=100, psize=100,  pc=1.0, nvm=1, clones=False,
        function="KP_p-2_n-20_ins-1", vhigh=0.1)
```