**BKnapsack.Console**

A .NET console application that executes experiments for solving the multiple knapsack problem using Binary Flower Pollination.

I'll attempt to explain the aspects of using this application. Read Carefully!

**Command Line Interface**

This application is to be executed from the command line. This is obtained with the CMD or COMMAND command in Windows OS.

Steps to execute:

* Download the [zipped file](https://github.com/mykeels/BKnapsack/blob/master/BKnapsack.Console/BKnapsack-Console.zip)
* Extract the zipped file into a directory
* Within that directory, you'll see Bknapsack.Console.exe ... [see example](http://take.ms/VJMdI)
* Click on the address bar and type cmd ... Press ENTER ... [see example](http://take.ms/rnUUh)
* A command line interface should pop up ... [see example](http://take.ms/dGNcN)
* Type BKnapsack.Console -d 1 1 -c config/config.json and press ENTER ... [see example](http://take.ms/EZB2z)

**Multi-Knapsack problem**

The multi-knapsack problem deals with selecting items of various values and weights into fictional knapsacks, making sure that we maximize the total value of items selected, and the total weight of items selected do not exceed the maximum weights of the knapsacks. For more, [read here](https://github.com/mykeels/BKnapsack/blob/master/BKnapsack.Console/Knapsack.md).

**Datasets**

Let's talk about datasets. Read more [here](https://github.com/mykeels/BKnapsack/blob/master/BKnapsack.Console/data).

**How to use**

The Command Line for this application makes use of flags, which are used to control the program. With flags, you can dictate actions like what experiments to run, how many times to run the experiments, what parameter values to pass to the algorithms used for the experiments, etc.

Flags available include:

**Data Flag**

With this flag, you can tell the application which dataset to use. The datasets are stored in the /data folder. You should see 9 folders, each containing 30 files. [See example here](http://take.ms/cogng).

You can choose for the program to work on one of these files at a time in two ways:

* Use the file path e.g. -d data/mknapcb1/mknapcb1-1.json
* Use the folder index and file index e.g. -d 1 1 which would select the first folder and first file in it.

**Config Flag**

The program uses a default configuration which is

{

"consoleWriteInterval": 100,

"enforceHardObjective": true,

"noOfIterations": 50000,

"populationSize": 5,

"writeToConsole": true

}

To configure it any differently, you need the config flag which is -c config/config.json which points to a config.json file where the program should get its alternate configuration from.

**Setup Flag**

When used from the command line, the program can only run one experiment at a time. If you want to run multiple experiments, use a setup.txt file, then point the program to it using the flag -s setup/setup.txt.

In the setup.txt file, you can define experiment setups one line at a time, such as:

-d 1 1 -r 3 -c config/config.json -levy 0.05

-d 1 2 -r 3 -c config/config.json

-d 1 3 -r 3 -c config/config.json

Note how in the code above, you can combine multiple flags. Enjoy!

**No-of-Runs Flag**

You may want to execute an Experiment a certain number of times. This is achieved using the -r flag, which is used as -r 3to execute the experiment 3 times. Change the number to any number you wish.

**Alpha Flag**

The Binary Flower Pollination Algorithm takes in a custom parameter which is called alpha, which takes a decimal value between 0 and 1 and is used to control Levy Flights. You can specify this using the -a flag, which is used as -a 0.3 for an alpha value of 0.3.

**Beta Flag**

The Binary Flower Pollination Algorithm also takes in a custom parameter which is called beta, which takes a decimal value which may be greater than 1. You can specify this using the -b or -beta flag, which is used as -b 0.1 for a beta value of 0.1.

**Population Size Flag**

The Binary Flower Pollination Algorithm also requires a population of flowers of a specified size. By default, the population size is 5, but it can be changed with the -size flag, which is used as -size 15 for a population size of 15.

Note: Experiment Results are stored in the `results` folder in JSON files, and are named in the following format: `{data-name}-{data-index}\_r{run-index}-file{file-index}.json` e.g. `mknapcb1-1\_r1-file3.json`

Have fun experimenting!