

iRace - MATLAB

Amirhossein Moosavi
email: amir.moosavi@uottawa.ca

Note: I assume that you are already familiar with different components of the iRace algorithm.

Introduction

In this document, I would like to briefly discuss running iRace for algorithms coded in MATLAB. To facilitate this connection, I use Python to code the target runner function. Thus, iRace (R) passes the input information to the target runner function (Python.) Then, the target runner function passes the input information to the main algorithm (MATLAB,) receives the outcome, and sends them back to iRace.

Connecting Python & MATLAB

As mentioned, I use Python to code the target runner function. To call a MATLAB session from Python, I use the MATLAB Engine API, which provides a package for Python to call MATLAB as a computational engine. The target runner coded in Python has a structure like below:

```
#!/usr/bin/python3

# ...
import matlab.engine
eng = matlab.engine.start_matlab()

# ...

# Get the parameters as command line arguments.
configuration_id = sys.argv[1]
instance_id = sys.argv[2]
seed = sys.argv[3]
instance = float(sys.argv[4])
cand_params = sys.argv[5:]

# Parse parameters
```

```

while cand_params:
    # Get and remove first and second elements.
    param = cand_params.pop(0)
    value = cand_params.pop(0)
    if param == "--MaxIt":
        MaxIt = float(value)
    elif param == "--nPop":
        nPop = float(value)
    #...

#...

# Run and print the output
print(str(eng.YourMainAlgorithm(instance, MaxIt, nPop, ...)) + '\n')
sys.exit(0)

```

Your main file in MATLAB should be a function that receives some input values (parameters of the algorithm and instance number) and run the main algorithm. Note that I usually pass instance number (a float) to the target runner in Python, and then, to the main file in MATLAB. Finally, MATLAB uses this number to read the input values of the test instance associated with the given instance number. Therefore, the structure of the main file in MATLAB looks like below:

```

function Cost = GeneticAlgorithm(ins, MaxIt, nPop, ...)
    % This code is for a minimization problem
    % ins => Instance number
    % MaxIt => Maximum Number of Iterations
    % nPop => Population Size
    % ...

    %% Problem Definition
    % Define the test instance here based on the given "ins"

    %% Initialization
    % Create the initial population, and evaluate and sort them

    %% Main Loop
    % For the given "MaxIt", apply the crossover and mutation operators
    % to generate new off-springs, evaluate them, and create the new
    % population

    % Return "Cost"
end

```

To be able to use the MATLAB Engine for Python, you should first check the compatibility of Python and MATLAB installed on your computer. Versions

of Python compatible with MATLAB (to use MATLAB Engine for Python) are listed in the last section [2]. Given the version of MATLAB on your computer, you should install the appropriate version of Python [7]. You should also make sure that the default version of Python on your computer is the one needed by MATLAB [8, 9].

You can find other useful links to configure your system for connecting Python and MATLAB in the last section [1, 3, 4, 5, 6].

Note: You must put your main algorithm and its component in a folder that has been added to the MATLAB search path.

Example

Suppose a very simple Genetic Algorithm (GA) that aims to solve an optimization problem (minimization.) For GA to work, we need to define a few input parameters (including their ranges) as follows:

- *ins*: Instance number (10, 20, 50, 100, 200, 400, 600, 800, 1000)
- *MaxIt*: Maximum Number of Iterations (type: c - values: (10, 20, 30, 40, 50, 100, 200))
- *nPop*: Population Size (type: c - values: (5, 10, 20, 30, 40, 50))
- *pc*: Crossover Percentage (type: r - values: (0.1, 0.8))
- *pm*: Mutation Percentage (type: r - values: (0.05, 0.5))
- *mu*: Mutation Rate (type: r - values: (0.01, 0.2))
- *beta*: Selection Pressure (type: c - values: (1, 2, 4, 6, 8, 10))

To tune these parameters, I have prepared a set of codes that connect iRace (in R,) the target runner function (in Python,) and the main algorithm (in MATLAB.) You could find this set of codes in the attachment. Furthermore, I have prepared a video that illustrates how to run these codes on Linux.

Hopefully, this brief tutorial is beneficial for MATLAB users.

Useful links

1. [MathWorks: Configure Your System to Use Python](#)
2. [MathWorks: Versions of Python Compatible with MATLAB Products by Release](#)
3. [MathWorks: Connect Python to Running MATLAB Session](#)
4. [MathWorks: Calling MATLAB from Python](#)
5. [MathWorks: How do I launch MATLAB on Linux?](#)

6. [MathWorks: Start MATLAB on Linux Platforms](#)
7. [Python.org: Download the latest releases](#)
8. [THEQUICKBLOG: How to change default version of Python as Python3?](#)
9. [StackExchange: Change the Python3 default version in Ubuntu](#)