Using HyFlex and irace for Automatic Configuration of Hyper-heuristics: A Quick Start

Weiyao Meng

This document serves as a quick-start guide for setting up the environment, executing the provided hyper-heuristics within HyFlex and utilising irace to configure the parameters of the provided hyper-heuristics. For comprehensive instructions and detailed explanations, please refer to the 'Detailed Instructions' document.

Setup

Setting up HyFlex

HyFlex can either be run using Java from the command line or by importing the files into a Java IDE. For simplicity, we use the Eclipse in this instruction.

1. **Installing Java:** https://www.java.com/en/download/help/download_options.html
2. **Installing Eclipse:** https://www.eclipse.org/downloads/
3. **Importing resources in 'project-hyflex' into Eclipse**
   - In Eclipse, navigate to **File** → **Import** → **General** → **Existing Projects into Workspace**.
4. **Adding HyFlex JAR file to classpath**
   - In Eclipse, right-click the project folder in the **Project Explorer** on the left side of the screen.
   - Select **Build Path** → **Add External Archives**.
   - Navigate to the location of the HyFlex JAR file (`chesc-fixed-no-ps.jar`) and select it.
   - Click **Open** to add the JAR file to your project's classpath.
5. **Verifying HyFlex is set up correctly**
   - In Eclipse, find the example program `examples.example.ExampleRun1.java` in your project.
   - Compile and run `ExampleRun1`.

If HyFlex is setup properly, you will observe the expected output in the console:

```
Algorithm: Example Hyper Heuristic One
Problem instance: class travelingSalesmanProblem.TSP
Time limit set to: 60 seconds
Search ...
```

You have attempted to run the `ExampleRun1` class. In this class, you will see that we create a Hyper-Heuristic object of the `ExampleHyperHeuristic1` type. The `Example Hyper Heuristic One` algorithm is executed on a specific TSP instance for 60 seconds. Once the termination time is reached, the console will display the best solution value found.

Setting up irace

Follow the **Setup** instructions in https://lopez-ibanez.eu/sigevo-summer-school-2023/.
For simplicity, we use the Rstudio console for the rest of the instruction.

Preparing Hyper-heuristics Runner for irace

To enable irace to execute the hyper-heuristics within HyFlex and configure parameters effectively, we need to ensure that the hyper-heuristics Runner is executable from the command line and capable of receiving parameters as arguments.

1. Prepare the Runner that can accept parameters passed as command-line arguments.
2. Exported the Runner as a runnable JAR file.
3. Move the JAR file into the designated folder.

Navigate to the 'examples.rn' package to find `RN`, which extends the `ExampleHyperHeuristic1` by allowing for the passage of 'i' and 'd' parameters.

The `RNRunner` executes `RN` and allows arguments (such as parameters, random seeds, instance ID and run time) to be passed from the command line. It returns the objective function value of the best-found solution. The corresponding JAR file is located in the designated folder.

Preparing Configuration Scenario in irace

The irace User Guide provides a guide for setting up a basic execution of irace in Section 4.1. The template files provided within the irace package serve as the starting point for creating new configuration scenarios.

In this challenge, example configuration files are provided to configure the heuristics of the hyper-heuristics `RN`. These configuration files, along with `RNRunner`, are located in the "project-irace/irace-hyflex-rn" folder.

Executing irace

The configuration scenario for the provided `RNHH` is now ready to use by irace. To execute the configuration scenario in the RStudio console, follow these steps:

1. Change the working directory to the 'irace-hyflex-rn' folder. If the folder's location is '/path/to/project-irace/irace-hyflex-rn', use the command:

```
setwd("/path/to/project-irace/irace-hyflex-rn/")
```

2. Load irace:

```
library("irace")
```

3. Run irace:

```
irace.cmdline("")
```

Congratulations! You've successfully implemented automatic configuration for `RNHH`.