

Evolutionary_Computation_HT2024

Table of Contents

- [Prerequisites](#)
- [Usage](#)
- [Authors](#)

Prerequisites

Make sure you have Python and pip installed on your machine.

1. Install Python

- **Windows:** Download the installer from the [official Python website](#) and run it. Make sure to check the box that says "Add Python to PATH" during installation.
- **macOS:** You can use Homebrew to install Python. Open your terminal and run:

```
brew install python
```

- **Linux:** Use your package manager to install Python. For example, on Ubuntu, run:

```
sudo apt update  
sudo apt install python3 python3-pip
```

2. Verify the installation

After installation, verify that Python and pip are installed correctly by running the following commands in your terminal or command prompt:

```
python --version  
pip --version
```

3. Clone the repository

```
git clone https://github.com/mosmar99/Evolutionary_Computation_HT2024.git  
cd Evolutionary_Computation_HT2024
```

4. Create a virtual environment (optional but recommended)

```
python -m venv venv
# Activate the virtual environment
# On Windows
venv\Scripts\activate
# On macOS/Linux
source venv/bin/activate
```

5. Install dependencies

```
pip install -r requirements.txt
```

Usage

You can view the sample logs found in the **sample logs** sub folder, these can be visualized by running (warning this will open 5 windows):

```
python parts/visuals.py
```

You can generate your own logs by running the different main files found in the **parts** sub folder. Below are the available main files and their respective functions, visuals will be stored in **logs** subfolder:

- **main_final.py**: Runs the the best version of the genetic algorithm, and visualises the results as a lineplot.
- **main_basic.py**: The most basic setup of our genetic algorithm, no visuals besides prints to the terminal.
- **main_setup_comparison_bar_chart.py**: Caution!, this will run multiple setups at all available cores on your pc. Will store the setups in **LHS_Setups.log**, will store the evaluations of the setups in **LHS_Setups_evals.log**
- **main_setup_comparison_density_heatmap.py**: Caution!, this will run multiple setups at all available cores on your pc. Will store the setups in **test_setups.log**, will store the evaluations of the setups in **test_results.log**
- **main_stagnation_mean_vs_max.py**: This file compares the genetic algorithm with regards to average / mean fitness score and maximum fitness score. creates a two-column lineplot visualisation of the results. Stores the data in **max_vs_mean_geno.log**
- **main_static_vs_dynamic_stagnation.py**: compares the static and dynamic stagnation handling, visualizes the results as a two-column lineplot. Stores the data in **dyn_vs_static_stagn.log**
- **main_with_and_without_genocide(static).py**: used to run the Genetic Algorithm with and without genocide, and create a two-column lineplot, Stores the data in **geno_plot8.log**
- **main_with_static_genocide_comparing_dynamic_vs_static_mutation_and_recombination_rates.py**: Runs the genetic algorithm, with static genocide and compares the algorithm with regards to dynamic vs static mutation and recombination rates. Visualizes the results in a lineplot. Stores the data in **dynamic_vs_static.log**

You can visualize the logs directly by running the following command:

```
python parts/visualise.py
```

Authors

- **Mahmut Osmanovic (mosmar99)**
- **Isac Paulsson (isacpaulsson)**
- **Sebastian Tuura (tuura01)**
- **Emil Wagman (Neobyte01)**
- **Mohamad AlKhaled (MohamadAlkhaled)**