Manual for Genetic Algorithm (GA) and Random Search Configuration Tuning Tool

Dylan Williams
Student ID: 2521133

Overview

This tool is designed for automatically tuning configurations of software systems within a constrained evaluation budget. It supports two methods:

- **Random Search (RS)**: baseline method for randomly selecting configurations.
- **Genetic Algorithm (GA)**: intelligent search method based on evolutionary algorithms.

Prerequisites

- Python 3.8 or later installed
- Dependencies installed from `requirements.txt`

Install Dependencies

Run the following command from your project root directory:

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

Directory Structure

- `datasets/` : Input CSV datasets for different systems
- `GA_RawRunData/` : Results produced by GA
- `RS_RawRunData/` : Results produced by Random Search
- `GAsearch_results/` : Individual GA run results
- `search_results/` : Individual RS run results
- `GA.py` : Genetic Algorithm implementation
- `RandomSearch.py` : Random Search implementation

Usage

Run Random Search

```
python RandomSearch.py
```

Generates result CSV files under the folder `search_results`.

Run Genetic Algorithm

```
python GA.py
```

Generates result CSV files under the folder `GAsearch_results`.

Run Multiple Tests (for statistical analysis)

- Genetic Algorithm multiple runs:

  ```
  python testGA.py
  ```

- Random Search multiple runs:

  ```
  python testRS.py
  ```

Run Statistical Comparison (Welch's t-test)

```
python t-test.py
```

Generates a CSV file named `Welch_ttest_results.csv`, comparing GA and RS performance per system.

Output

- Best fitness scores are stored in `GA_RawRunData/` and `RS_RawRunData/`
- Summary statistics (mean, SD) are computed automatically
- T-test results are saved to `Welch_ttest_results.csv` in the project root