EA Practical Assignment (GA + ES)

This project contains:

- A Genetic Algorithm (GA) for PBO F18 (LABS) and F23 (N-Queens)
- An Evolution Strategy (ES) for BBOB F23 (Katsuura, d = 10)
- Plotting scripts to analyse performance across dimensions and strategies.

The tuning scripts for GA and ES each respect a hard budget of **100,000 function evaluations in total** across their tasks.

---

1. Setup

```
python3 -m venv venv
source venv/bin/activate        # Windows: venv\Scripts\activate
pip install -r requirements.txt # ioh, numpy, matplotlib, etc.
```

Make sure the `ioh` package is installed so the PBO/BBOB problems and IOH logging work.

---

2. Tuning the GA (Part 1)

Hyper-parameter tuning for GA on:

- F18 LABS (dim = 50)
- F23 N-Queens (dim = 64)

The script searches GA parameters under a joint tuning budget of **100,000 evaluations** and writes the best configuration to `ga_best_params.json`.

```
python s4822285_s4714067_tuning.py
```

After running:

- Check the console log for Stage 1 and Stage 2 statistics.
- The selected parameters are saved in `ga_best_params.json`, which are then used automatically by the GA implementation.

---

3. Running the tuned GA across dimensions

The GA script should be able to:

- Use the parameters from `ga_best_params.json`.
- Run 20 independent runs for each selected dimension and log to IOH format.

```
python s4822285_s4714067_GA.py --multi-dim
```

This should generate IOH logs under a `data/` directory (or similar), with `IOHprofiler_*.json` and `.dat` files for each problem and dimension.

---

4. Analysing GA performance vs dimension

Use `plot_dim_ga.py` to aggregate runs per problem and dimension and to create plots:

```
python plot_dim_ga.py \
  --data-root data \
  --out plots_ga_dims \
  --algo-prefix s4822285_s4714067_GA \
  --budget 5000
```

This produces, for each problem:

- Overlay of mean best-so-far curves vs evaluation fraction
- Heatmap of mean best-so-far (dimension × budget fraction)
- Final performance vs dimension (mean ± std + best-found star)
- AUC vs dimension
- Final boxplots of best-so-far across dimensions
- For N-Queens, success-rate vs dimension

The plots are written into `plots_ga_dims/`.

---

5. Running Evolution Strategies on Katsuura (Part 2)

The ES script compares classic (μ, λ) and $(\mu + \lambda)$ strategies on:

- F23 Katsuura (BBOB), dimension 10
- Budget: 50,000 evaluations per run
- Runs: 20 per configuration

Single baseline strategy

```
python s4822285_s4714067_ES_strategies.py
```

This runs the default configuration (e.g., `(1,10)` ES) and logs IOH data under `data/ES_STRATEGIES/`.

Comparison suite (multiple strategies)

```
python s4822285_s4714067_ES_strategies.py --compare
```

This runs a predefined set of (μ, λ) pairs (e.g., `1:10`, `1:30`, `3:30`, `5:50`, `10:50`) in comma and/or plus modes, each with 20 runs and the given budget, and writes IOH logs for each configuration.

You can also provide your own suite:

```
python s4822285_s4714067_ES_strategies.py --pairs 1:10 3:30 5:50 --both
```

---

6. Analysing ES strategies on Katsuura

After running ES, use your ES plotting script to produce:

- Mean best-so-far curves per strategy vs evaluation fraction
- Final performance (mean ± std) vs strategy
- AUC (search efficiency) vs strategy
- Boxplots of final best-so-far per strategy

Since the Katsuura summary JSON is already generated (`F23_Katsuura_d10_summary.json`), you can also load it directly in notebooks or scripts to reproduce the final-performance and AUC plots.
