# 01_scaling_01_run_experiment

June 25, 2024

## 1 Scaling - running experiment

Runs a series of experiments of OpenTTDLab, varying the number of workers used, what data is retrieved from the worker processes, recording how long each takes.

Results are saved to 01_scaling_raw.csv.

```
[ ]: !python -m pip install OpenTTDLab==0.0.72 pandas==2.2.0
```

```python
[ ]: import time
from openttdlab import run_experiments, bananas_ai

def all_results(row):
    return [row]

def minimal_results(row):
    # It is expected early on the game that no company value is recorded
    def get_company_value(player):
        try:
            return player['old_economy'][0]['company_value']
        except (KeyError, IndexError):
            return 0

    return [{
        'date': row['date'],
        'company_value': get_company_value(row['chunks']['PLYR']['0']),
        'error': row['error'],
    }]

def run_experiment_timed(max_workers, result_processor):
    start = time.monotonic()
    results = run_experiments(
        openttd_version='13.4',
        opengfx_version='7.1',
        experiments=(
          {
            'seed': seed,
            'ais': (
```

```
                    # This is trAIns 2.1. The md5 enforces a specific version
                    # of trAIns and was retrieved by a previous call to
                    # openttdlab.dowload_from_bananas
                    bananas_ai(
                        '54524149', 'trAIns',
                        md5='c4c069dc797674e545411b59867ad0c2',
                    ),
                ),
                'days': 366 * 4 + 1,
            }
            for seed in range(0, 50)
        ),
        result_processor=result_processor,
        max_workers=max_workers,
    )
    assert all(not row['error'] for row in results)
    end = time.monotonic()
    return (end - start)

results = [
    {
        'max_workers': max_workers,
        'results': 'all',
        'wallclock_time': run_experiment_timed(max_workers, all_results),
    }
    for max_workers in range(1, 9)
] + [
        {
        'max_workers': max_workers,
        'results': 'minimal',
        'wallclock_time': run_experiment_timed(max_workers, minimal_results),
    }
    for max_workers in range(1, 9)
]
```

```
[ ]: import pandas as pd

     df = pd.DataFrame(results)
     df.to_csv('01_scaling_results_01_raw.csv', index=False)
```