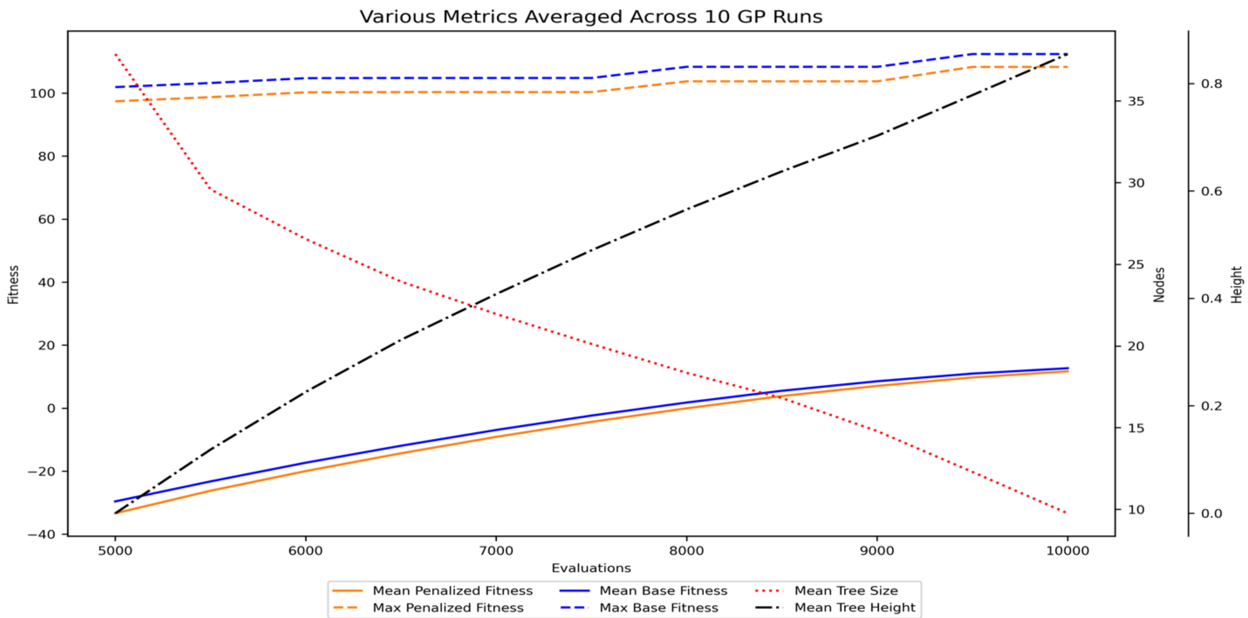
Mohab Yousef.
COMP 5660 Fall 2024 Assignment 2b
Mey0012

Green_Config: used high mu, low parsimony coefficient, and mid num of children to get better results but it took longer time.

```
[ea]
mu = 5000
num_children = 500
mutation_rate = 0.05
parent_selection = k_tournament_with_replacement
survival_selection = k_tournament_without_replacement
individual_class = TreeGenotype

[parent_selection_kwargs]
k = 3

[survival_selection_kwargs]
k = 5

[fitness_kwargs]
parsimony_coefficient = 1/7
experiment = green

# Make sure these are the same as the [pac_init] header in your 2a experiment.
[problem]
depth_limit = 7
terminals = ('G', 'P', 'F', 'W', 'C')
nonterminals = ('+', '-', '*', '/', 'RAND')

[recombination_kwargs]
depth_limit = ${problem:depth_limit}
terminals = ${problem:terminals}
nonterminals = ${problem:nonterminals}

[mutation_kwargs]
depth_limit = ${problem:depth_limit}
terminals = ${problem:terminals}
nonterminals = ${problem:nonterminals}

# Don't touch any of these.
[game]
game_map = map.txt
pill_spawn = waves
pill_density = 1/3
fruit_prob = 1/75
fruit_score = 10
fruit_spawn = corners_spawned
time_multiplier = 2.5
num_pacs = 1
num_ghosts = 3
```



Mean Penalized and Mean Base Fitness started low at a negative value then steadily improved which means better solutions are being generated as the the algorithm progresses, and the GP is learning and creating better solutions over time.

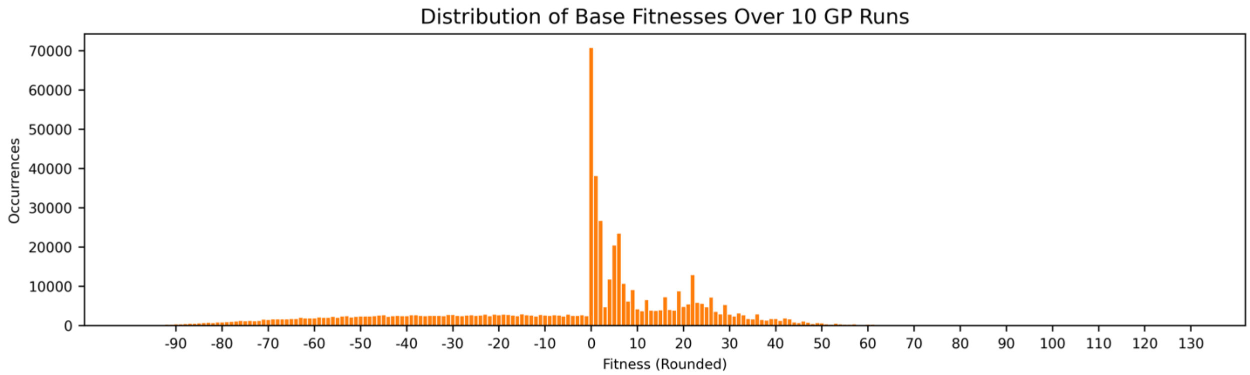
Max Penalized Fitness and Max base fitness started high then leveled off which means the algorithm did find a good solution but it started improving slowely as the algorithm continued.

Mean tree size decreases initially showing that the algorithm is pruning complex solutions early on to improve efficiency, as the evaluations progress the tree size stabilizes and slightly increases which means the algorithm starts allowing more complexity as it converges.

Mean Tree Hight steadily increases throughout the experiment showing that while trees grow in complexity they are still manageable in size, and that the solutions are becoming more complex over time

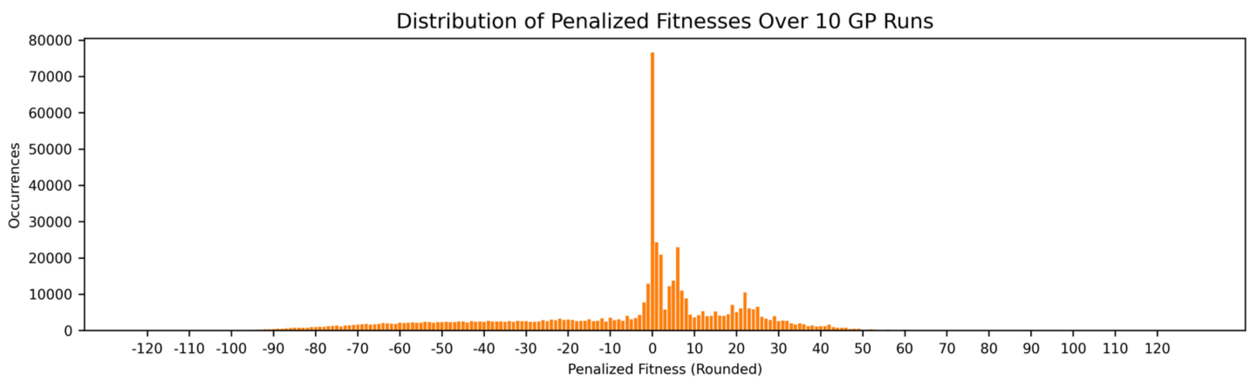
There's a balance between tree size (early pruning) and controlled growth (increasing height) which meant that the algorithm avoids bloat while still allowing for complexity where necessary.
Overall: the plot shows that the GP algorithm is working well, it's improving solution quality while managing complexity and reaching a good balance between simplicity and performance.

Base Fitness scored over 10 runs.



Most solutions had base fitness scores around 0 which means solutions were neither very good nor very bad, and a few solutions had higher fitness (above 40), showing the algorithm can find good solutions but not as many, and Some solutions had negative scores, indicating poor performance.

Penalized fitness distribution over 10 runs.



Penalized fitness scores are same the base fitness around (0) but many scores are much lower (below -100) due to penalties for complexity, and there is fewer high scores compared to base fitness which means penalties discourage overly complex solutions.

Overall: the algorithm finds some great solutions, but it mostly generates average ones.

Statical Analysis:

```python
[1]: from stats import run_stats
# TODO: Conduct statistical analysis comparing the results of 2a and 2b
from stats import run_stats

# File paths for datasets
dataset1 = './data/2a/green/best_per_run.txt'
dataset2 = './data/2b/green/best_per_run.txt'

# Run statistical analysis
run_stats(dataset1, dataset2)
```

```
Number of samples: 10
./data/2a/green/best_per_run.txt mean: 118.19215946843853
./data/2a/green/best_per_run.txt stdv: 10.171452003527722
./data/2b/green/best_per_run.txt mean: 112.4456699889258
./data/2b/green/best_per_run.txt stdv: 11.76821781276956
p-value: 0.2582567223313278
```

Assignment 2a (Green Experiment):
Mean best per-run fitness: 118.19
Standard deviation: 10.1
Assignment 2b (Green Experiment):
Mean best per-run fitness: 112.45
Standard deviation: 11.76
P-value: 0.258 (from statistical t-test)

The mean fitness in Assignment 2a is slightly higher than in Assignment 2b meaning the solutions in Assignment 2a generally performed slightly better**. (This might be not very accurate analyzation as I had to fix my parse tree for assignment 2a and run cells again)**
Standard Deviation: Both experiments have close scores which means consistent performance across runs.
The p-value of 0.258 indicates that the difference in mean fitness between the two experiments is not statistically significant.

Yellow_Configs

```
[ea]
mu = 1000
num_children = 500
mutation_rate = 0.05
parent_selection = k_tournament_with_replacement
survival_selection = k_tournament_without_replacement
individual_class = TreeGenotype

[parent_selection_kwargs]
k = 2

[survival_selection_kwargs]
k = 3

[fitness_kwargs]
experiment = yellow
parsimony_coefficient = 0   # Set to 0, as parsimony pressure is now a separate objective

# Make sure these are the same as the [pac_init] header in your 2a experiment.
[problem]
depth_limit = 7
terminals = ('G', 'P', 'F', 'W', 'C')
nonterminals = ('+', '-', '*', '/', 'RAND')

[recombination_kwargs]
depth_limit = ${problem:depth_limit}
terminals = ${problem:terminals}
nonterminals = ${problem:nonterminals}

[mutation_kwargs]
depth_limit = ${problem:depth_limit}
terminals = ${problem:terminals}
nonterminals = ${problem:nonterminals}
```

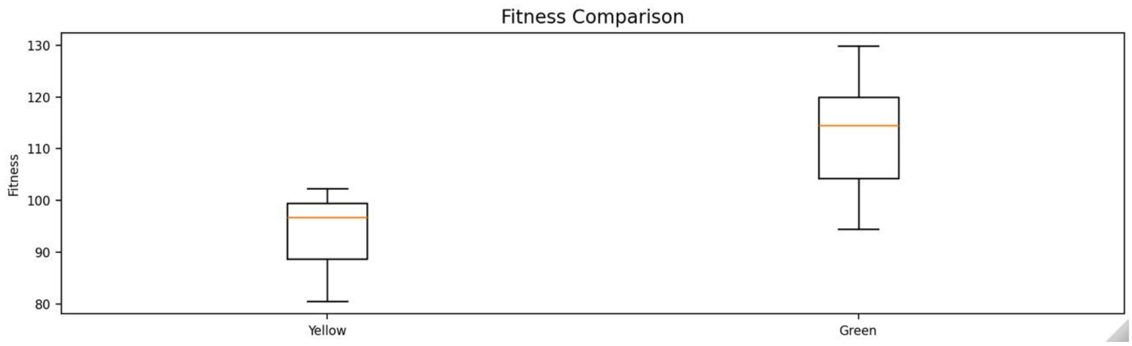
Dropped mu to 1000 for fast run and Parsimony_coeffiecnt to 0 because it is a sebrate object.

Did not really got time to generate the plots, tune, run 10 runs experiment but I saved the Global Pareto front, and compared the results I got from it with the results from 10 runs green.

```
plt.show()
```

```
Yellow Data:
Mean: 93.18877814691768, Std Dev: 11.296178568796272
Green Data:
Mean: 112.4456699889258, Std Dev: 11.76821781276956
T-statistic: -2.5645408227872157, P-value: 0.07244582323657829
```

Figure 1



The p-value of 0.0724 is above 0.05 meaning Green outperforms Yellow in terms of mean fitness
The mean fitness for the Green experiment is higher, indicating better overall performance compared to the Yellow experiment.
The box plot confirms the difference in central tendencies (median) between Yellow and Green experiments.

In gPac_population_evalution.py I implemented the base population evolution for yellow and reds form 1-5 because I tended to to them but ran out of time .

The author acknowledges the use of [ChatGPT -4] developed by [Open AI] in the preparation of this assignment. It was used in the following way(s): [code generation, fixing parse tree from assignment 2a, and fixing grammar In the report]