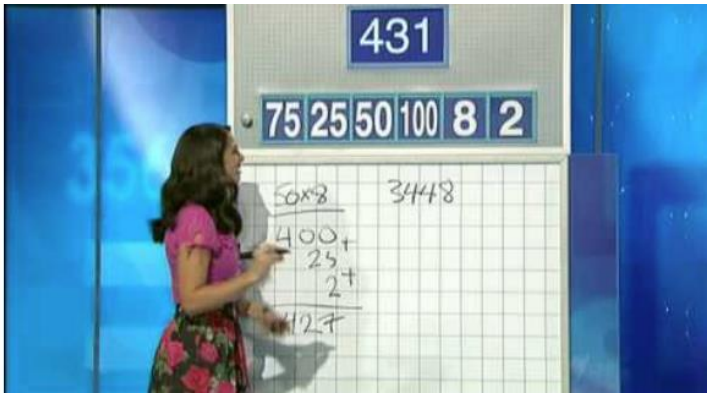
CAB320 Assignment 2

Letters and Numbers Assignment



Name	ID
Nihar Rupareliya	N10335243
Ricky Lau	N10330895
Cory Bullen	N10467114

Due: 6th June 2021 11:59 PM

Table of Contents

Introduction

The Letters and Number game is quite popular amongst different nationalities. The player is given a target value and must choose small values (between 1 to 10) and large values (25, 50, 75 or 100) up to six digits. They then are given 30 seconds to get as close to the target value as possible using the arithmetic operators. In the following experiment, we are using a genetic algorithm, a search heuristic that reflects the natural selection of fittest individuals to produce the upcoming offspring to evolve a population of expression trees. The fitness for us is determined by the distance between the target value and the value computed using the expression trees. Furthermore, we investigate the best trade-off between the population size and the maximum number of generations such that the process is most efficient.

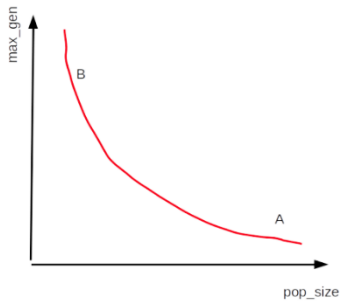


Figure 1 Relation between number of generation and the population size

(Figure taken from Assessment specification slide)

Methodology

We choose a fixed tree and the target value for our testing, which are "[100, 50, 3, 3, 10, 75]" and the "449", respectively. For getting the best trade-off value between the population size and the max generation, we will follow the steps below:

1. **Find the maximum population size for our given tree and the target value:**
   For the analysis part, we choose the full population size that we will evaluate to be 1000.

2. **Create a list of 20 population from the range of 5 to maximum population size (1000) and evaluate each of the generations for five different times:**
   We get the list of 20 different populations, and we run our program for 2 seconds to get the maximum generation that each of them can reach. After that, we run each of the population values five times, thus running the evaluation function 100 times.
   The next step is to sort the list and get the median value for each of the maximum generation that we reached for the particular population, which leaves us with 20 numbers of the maximum generation that each of the population can get in 2 seconds.

```
POP: [18, 21, 244, 250, 285, 294, 305, 345, 355, 363, 374, 452, 468, 583, 609, 637, 754,
861, 888, 957]
MAX GEN:  [1354, 987, 115, 89, 98, 82, 87, 80, 76, 72, 55, 53, 63, 44, 43, 32, 32, 5, 5, 4]
```

Figure 2 The population list and its respective maximum generations

As we can see the relationship from *Figure(1)* here, as the population size increases, the number of maximum generations decreases and vice versa.

3. **Find the success rate on the following population value and their maximum generation:**
   After getting the values of maximum generation for each population, to get a good idea about the success rate, we run the evaluation of the genetic algorithm 30 times and grab the cost of each one of them, which creates an extensive list of 30 success rate for each population size, i.e., 600 different values of success rates which we will use for further calculations.

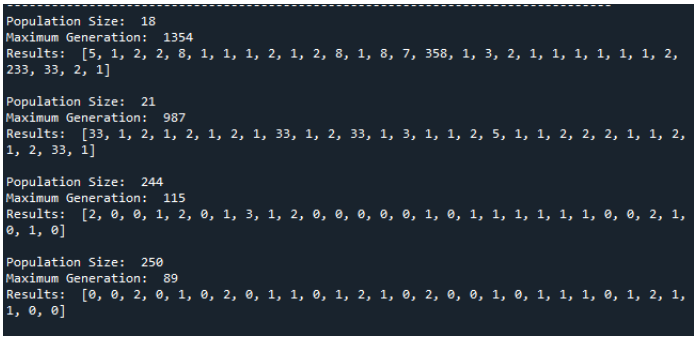


Figure 3 Success rates of the first four population size

4. **Calculate the number of times the success cost was zero for each of the population to get a relation between the population, the generation, and their success rate:**

We find the number of times our population has 0 costs at the end, or it is a perfect match, which gives us how many times was our algorithm was successful in getting up to the ideal target value.

Figure 4 List of the scores for the perfect score

5. **Find the pair of population and maximum generation that has the best payoff:**

We find the maximum number of success rate from the last step and associate the values of that to the maximum generation and the respective population size to get the best trade-off associated with our generation and population.

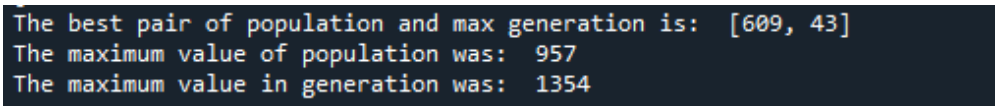


Figure 5 The best trade-off pair

We then generate the charts from the values produced from the methodology for evaluation.

Evaluation

The following tree was evaluated twice to see the credibility of the result and see if the results follow the general trend as seen clearly from *Figure (6)* and *Figure (7),* as we increase the population size or as the number of populations increases (iterations for the genetic algorithm), the chance of getting a perfect score increases and thus the success rate also increases.

However, there is a point at which the graph becomes saturated, and the value of success does not increase anymore and decreases.

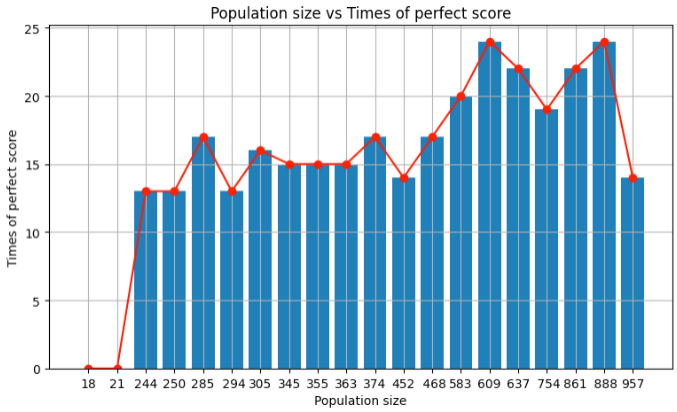


Figure 7 Population vs Times of perfect score with Result 1

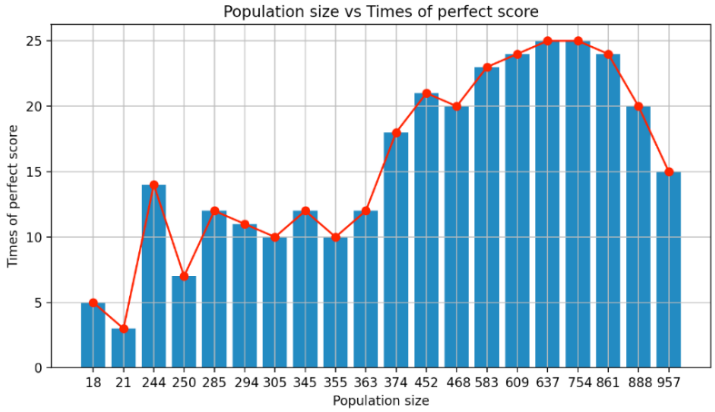


Figure 6 Population vs Times of perfect score with Result 2

For our purpose of Tree one, which is T = "[100, 50, 3, 3, 10, 75]" and the target value = "449",

During both experiments, we found that the best trade-off value to be sort of lying at around 609 population size generated a good number of perfect scores for our target value.

Conclusion

In conclusion, the method is feasible, and it produces a good result. As in evaluation, we found a trend of getting better trade-off when the population size increased within our range. Our approach also proves that the best trade-off would not be combined with the most diminutive population size as *Figure (1)* shows. The best trade-of value is hence found to be lying somewhere near to the shoulder of the *Figure 1*.

Future work

In the future, if our team could have the chance of working on this project again, we might want to expand the dataset, which could be the number and range of population size. At this stage, our method is showing the increment of accuracy of the data only. we might want a larger dataset to see the peak and downhill of the result. Therefore, we could visualize the best range of population size and their according maximum generation.

APPENDIX A

Statement Of Completion:

Task 1:

Function	Comment
polish_str_2_expr_tree	✓
decompose	✓
mutate_num(T, Q)	✓
mutate_op	✓
cross_over	✓

Table 1 Statement of Completion

Task 2:

Task	Comment
Evaluation	✓