

“Why Genetic Algorithms Are More Suited to Solve Certain Types of Optimization Problems”

An Honors Thesis

by

Leslie E. Peters

California, Pennsylvania

2020

California University of Pennsylvania

California, Pennsylvania

We hereby approve the Honors Thesis of

Leslie E. Peters

Candidate for the degree of Bachelor of Science

Date

Faculty

4-22-2020



Jennifer Wilburn, PhD
Honors Thesis Advisor

4-22-2020



Brenton Wilburn, PhD
Second Reader

4-22-20



Loring Prest, MS-LIS, ThM
Honors Advisory Board

4/22/20



Craig Fox, PhD
Associate Director, Honors Program

22 April 2020



M. G. Aune, PhD
Director, Honors Program

Acknowledgements

I would like to thank the Honors Program for the opportunity to work on this thesis project and to expand my knowledge. I would also like to thank my committee for their cooperation and help with this project; Dr. Brenton Wilburn, Dr. Jennifer Wilburn, and Prof. Loring Prest. A special thanks to Dr. Jennifer Wilburn for the patience and time she put into holding meetings and helping me understand certain aspects of this project.

Abstract

Some technical optimization problems that have been normally solved manually or with other traditional techniques, can be optimized by using different advanced techniques. Techniques have evolved to include artificial intelligence (AI) algorithms to get better results. Genetic algorithms are a great example of one of these optimization techniques that uses AI. While these techniques are designed to solve problems more efficiently than before, there are some problems that are better suited to these algorithms compared to others. By analyzing experimental data, it is possible to discern what types of problems are better solved using genetic algorithms are better at solving. Studies that have isolated genetic algorithmic applications have been chosen to better compare the solutions that were found, straying away from hybridizations of any kind that could influence results and blur the line of responsibility. Five optimization problems were analyzed to find the problems that had the best improved results and the style best suited for genetic algorithms. These optimization problems that were studied include: PID control tuning; traveling salesman shortest path; generation of machine turning code; design of a wind turbine tower; and A-Mazer, a maze-solving algorithm. By breaking down these problems and how genetic algorithms are used to enhance solutions, a comparison can be made. It was found that problems that would have multiple trials within traditional techniques are more suited for genetic algorithms over other optimization problems.

Table of Contents

Background	Error! Bookmark not defined.
PID Control	Error! Bookmark not defined.
Traveling Salesman	Error! Bookmark not defined.
Optimization of Machine Parameters for Tuning Operations	Error! Bookmark not defined.
Wind Turbine Tower	Error! Bookmark not defined.
A-Mazer	Error! Bookmark not defined.
Conclusions	Error! Bookmark not defined.
References	Error! Bookmark not defined.

Background

Genetic algorithms (GA) are numerical algorithms which are inspired by the biological process of natural selection. They are designed after the biological method of how genes are passed on throughout generations. Potential solutions to the problem represent individuals in the population. These are simply strings of data that indicate the key parameters to define a solution. A set of solution criteria behaves as the environment to define which solutions are better - "more fit". The evolutionary process consists of three primary processes: selection, crossover, and mutation. The selection process is where the fitness of each individual is evaluated. These strings of data progress over to the crossover stage, where individuals are chosen at random to produce new individuals composed to shared parts of both "parents". The resulting string is made from the selected pair; sharing data from both. The population will then enter the mutation phase where bits of that data will be randomly altered to create variety in the code. With this step, new possibilities may be considered that might never have shown up with only the selection and crossover stages. Some of these mutations may be beneficial, others may not. The better solutions will be selected to start the selection, crossover, and mutation process over again. These steps will be repeated until a convergence criterion is met or the simulation is manually stopped. At times, the engineer will not be satisfied with the results after numerous tries with the optimization and a new initial population will be constructed for re-optimization.

PID Control

One optimization problem that can be solved by genetic algorithms is PID control. Traditionally, PID control can be tuned by an engineer to get the desired results in a system response. While this is often an effective method, other methods were discovered to generate faster and more accurate results, such as the genetic algorithm. PID comes in three forms of systems;

Proportional(P), Proportional Integral (PI), and Proportional Integral Derivative (PID). PID systems have gains that need to be adjusted to create a stable system response. P systems have K_p gains, PI systems have K_p and K_i gains, and PID systems have K_p , K_i , and K_d gains.

A system response is the output of a system, which has already been given a desired state by the engineer. This desired point, the setpoint, is represented as a horizontal line across the output graph of the system. The overshoot and undershoot of the graph, as represented below in Figure 1, should be as small as possible for optimal results. The rise time, the time it takes to rise from 10% to 90% of the stability condition, indicates the speed of the response. This should also be minimized. The oscillations after reaching the setpoint should also remain bounded near the setpoint to maintain stability and can be measured as settling time. Shorter settling time indicates better response.

The way that a system response behaves is like a thermostat. The desired temperature would be the setpoint and once set, the temperature would try to reach that setpoint. The temperature might go higher than what was desired, which would be the overshoot. If it dips lower than the desired temperature, that would be the undershoot. The thermostat shuts off and turns back on to remain at the desired temperature, which would be the steadying oscillations that is measured as the settling time.

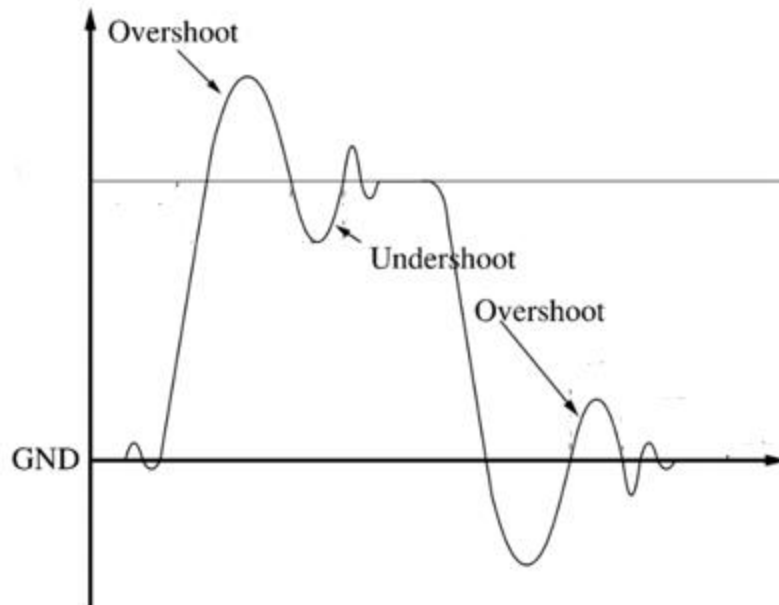


Figure 1: System Response Definitions

(Champak, Avendano, & Figueras, 2010)

For the gains in a system to be optimized, the values must be adjusted within the written code, which is then updated in the equation. The governing equation is split off into one, two, or three equations respectively for the different P, PI, and PID systems.

PID Original Equation:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Mjahed, Mostafa created a study on PID control with genetic algorithms to analyze the effects of before and after the implementation. All three gains; K_p , K_i , and K_d , were gathered in one string and implemented within the genetic algorithm's selection, crossover, and mutation

processes. By doing this, the solution that was optimized was composed of the gains. These gains were fed into the system to test the system response for each individual, at each iteration.

When only using traditional techniques, the results are as shown below in Figure 2. When using the genetic algorithm, the system response is as shown below in Figure 3. The overshoot had decreased by 14.7%, the settling time decreased by 137.7 seconds, and the rise time decreased by 43.3 seconds. This indicates that the algorithm was capable of finding a better solution than could have been found using traditional means.

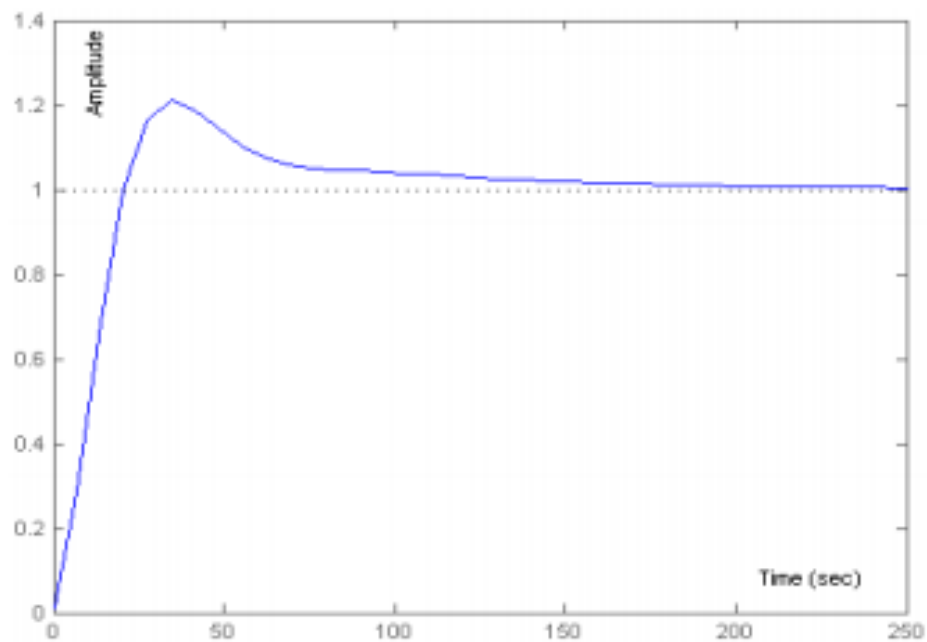


Figure 2: Traditional Technique System Response

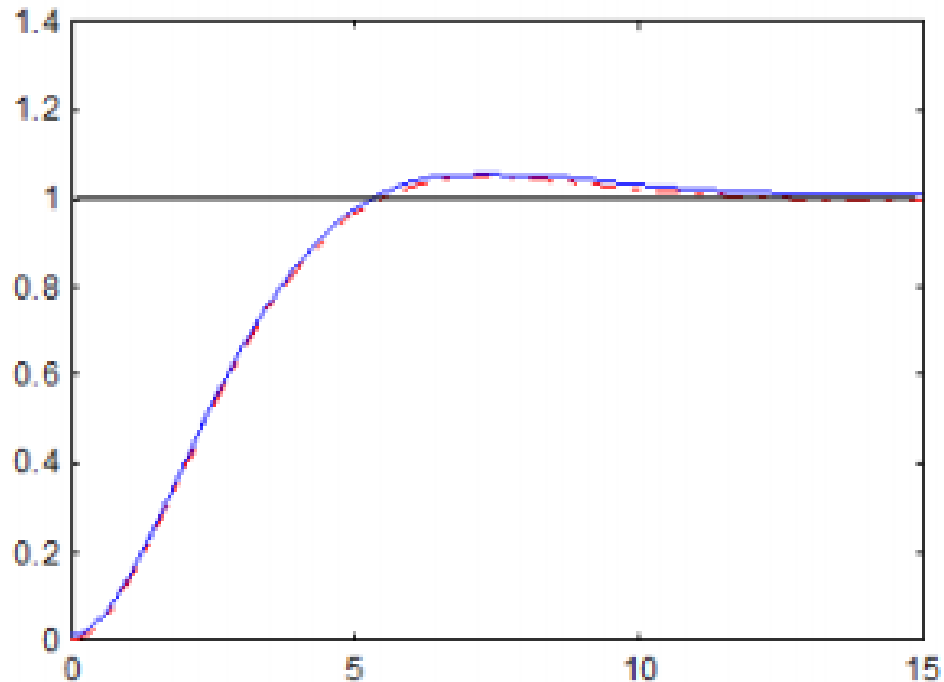


Figure 3: GA System Response

Traveling Salesman

For the traveling salesman optimization problem, the goal is different than PID control. In this problem, there are multiple locations which a *salesman* must visit. The best path taken to hit all these stops in the quickest fashion is the goal. This problem would normally need to be solved using “brute force” – checking all possible solutions. This is obviously cumbersome, so a method capable of finding the optimal solution without having to check all possible solutions is beneficial.

In this experiment by Hazim, Ziyad & Al-Jabbar, Abid, researchers looked at comparing the relative time and number of operations needed to solve the problem using the genetic algorithm versus traditional methods. Points of Duhok City were used as the locations, and the shortest path was to be found to hit all indications regardless of roads or other obstacles.

Excel was used to store the locations of Duhok City that were to be used to come up with possible paths to start the genetic algorithms process. Locations without a path for Duhok City Areas have the longitudes and latitudes shown in Figure 4. MATLAB served as the environment for the genetic algorithm.

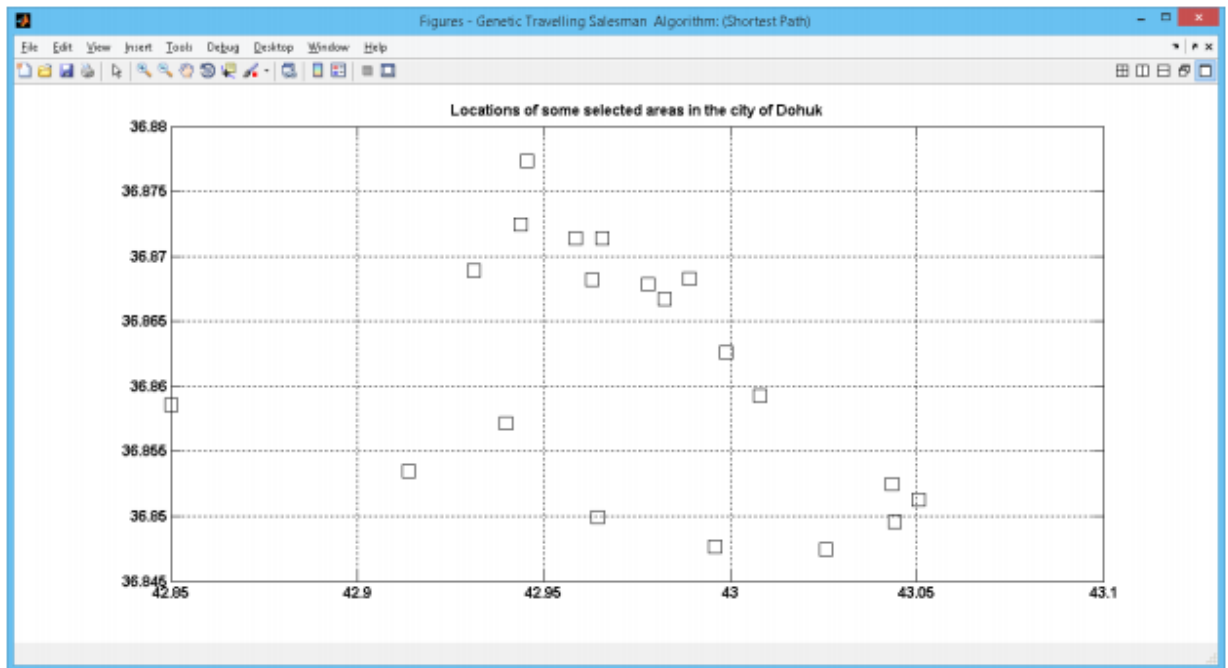


Figure 4: Selected Locations

Results for this genetic algorithm showed that the algorithm was capable of converging to the same global optimum solution for each trial, even though each trial started from a different population of solutions, and ultimately took a different number of generations to converge. Time to converge ranged from 2.49 seconds to 7.43 seconds – vastly shorter than a brute force approach, while yielding the equivalent result. The optimal solution obtained in each trial is shown in Figure 5 below.

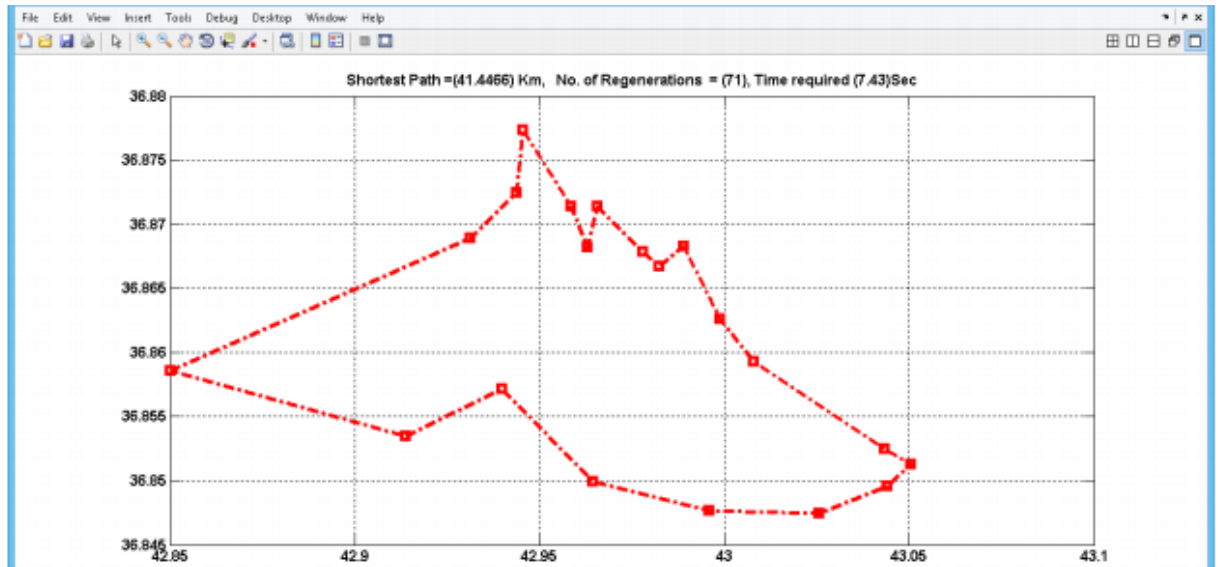


Figure 5: Optimal Path Result

Optimization of Machine Parameters for Tuning Operations

One common production machine for manufacturing is a lathe. A lathe is used to rotate material into which a cutting tool is driven to remove material. This is also called machine turning. The machines are often automated using Computer Numerical Cutting (CNC) technology. When performing CNC cutting, the machine must be programmed as to where and how to cut the material. How the material is cut is as important as what cuts to make. Proper tool positioning, selection of appropriate spindle speed, feed rate, and other cutting parameters have a strong impact on the quality of the finished part, the speed at which the part can be produced, and how long the tooling will last. These economic factors make proper selection of these parameters extremely important.

Traditionally, “rules of thumb” guidelines are used to select these parameters, based on the material being cut, the type of cut being performed, etc. If this is instead preformed using a genetic algorithm, a more efficient result can be achieved.

A series of constraints for the cutting parameters based on minimizing machining cost are mathematically derived and used within the genetic algorithm to provide the environment. MATLAB was used to solve as a programming base. The goal is to acquire more accurate boundaries for the cutting process in machining. This saves money, cuts down time, and allows the tools to last longer.

$$C = 0.3 + \frac{4.6}{V_c f} + 1.72 * 10^{-11} V_c^{4.55} f^{0.67}$$

The C stands for machining cost measured in euros and $V_c f$ are the constraints. By using these variables, an equation for machine turning can be found. The code for the genetic algorithm will enforce the constraints.

```
function C = turning_cost(x)
C = 0.3 +  $\frac{4.6}{x(1) * x(2)}$  + 1.72 * 10-11 * x(1)4.55 x(2)0.67;
end
```

The nonlinear constraints also need to be taken into consideration.

```
function [c, ceq] = nonlinear_constraints(x)
c = [x(1) * x(2)0.3-91.57;
x(1) * x(2)0.75-74.8;
x(2)0.75-6.48];
ceq = [];
end
```

Once this code is made, MATLAB can be used to execute and go through the genetic process to find the optimization solution.

The genetic algorithm takes 60 seconds to generate results for a total of 8 trials. The solution created was $C_{min}=0.335519$ (euros), $vc=14.395$ m/min and $f=9.0$ mm/rev.

Wind Turbine Tower

Genetic algorithms can also be used to reduce the size of the wind turbine towers. This is valued because it, in turn, lowers cost. Shigeo Yoshida experimented to create an optimization program from the genetic algorithm which determines the diameter of the whole tower, thickness of the wall, and position of flanges and doors to the navigation lights when facing certain factors.

When creating a wind turbine tower, many factors and guidelines are made that interfere with the design and make of the tower. To determine new guidelines to go by, using the genetic algorithm, there are a few steps that had to be taken. The materials, structure, Building Standard Law of Japan (BSL) force conditions, general guidelines, genetic algorithm limits, and other needs had to be considered along with the calculated load that was determined from the original tower and the aeroelastic. From this, IEC61400-1/Germanischer Lloyd (IEC/GL) forces from the design and safety variables were found.

These can then find the elements of the optimization program, which includes multiple factors of the tower. The genetic algorithm uses a solution domain that has been converted to biological terms and a fitness equation to use on this domain. This equation, along with modal analysis for 1st to 3rd bending forms, BSL forces, stability, fatigue damage, diameter distribution, and location of flanges and doors to navigation lights are what are considered for the optimization program and help with the design. The program can now find the necessary limits for the new tower.

Shown in Figure 6, is the relationship that the progress of iterations in the genetic algorithm had on fitness.

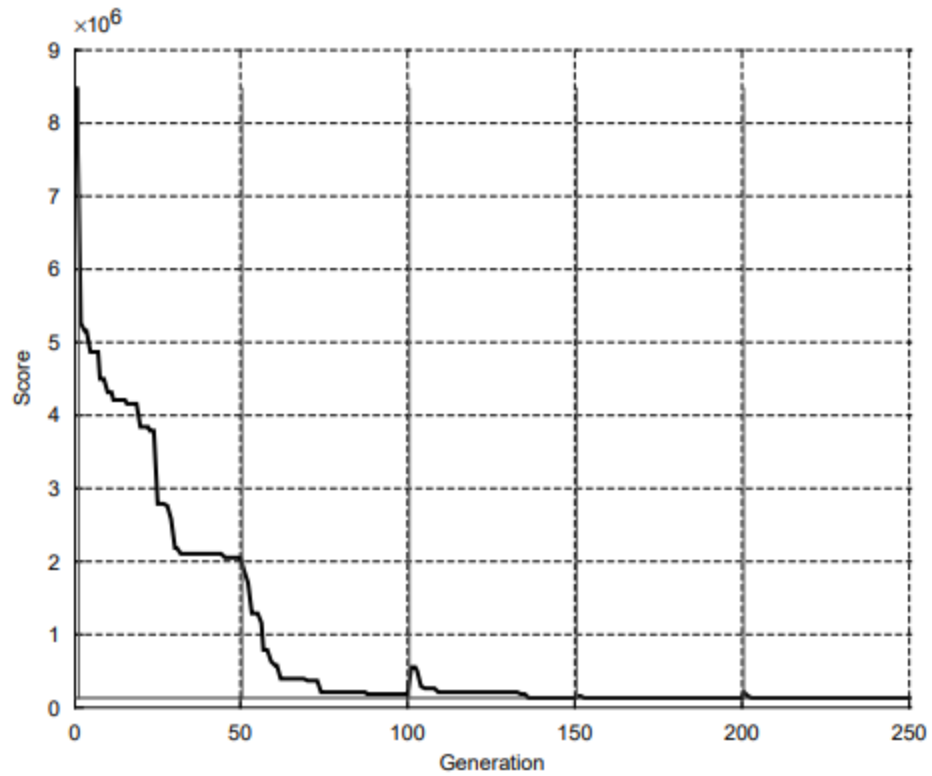


Figure 6: Performance of Model per Generation

The genetic algorithm was able to take different parameters in the tower and adjust to find the optimal results for each specific variation. Tower A; as shown in Figure 7, and Tower G; in Figure 8, are the first and last results respectively. Tower A through Tower G have different parameters.

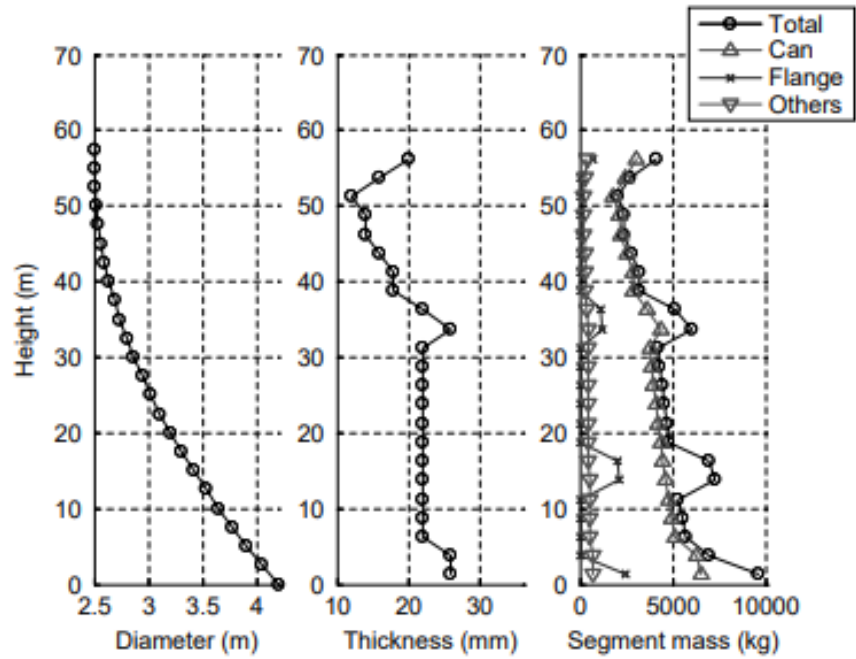


Figure 7: Tower A Results

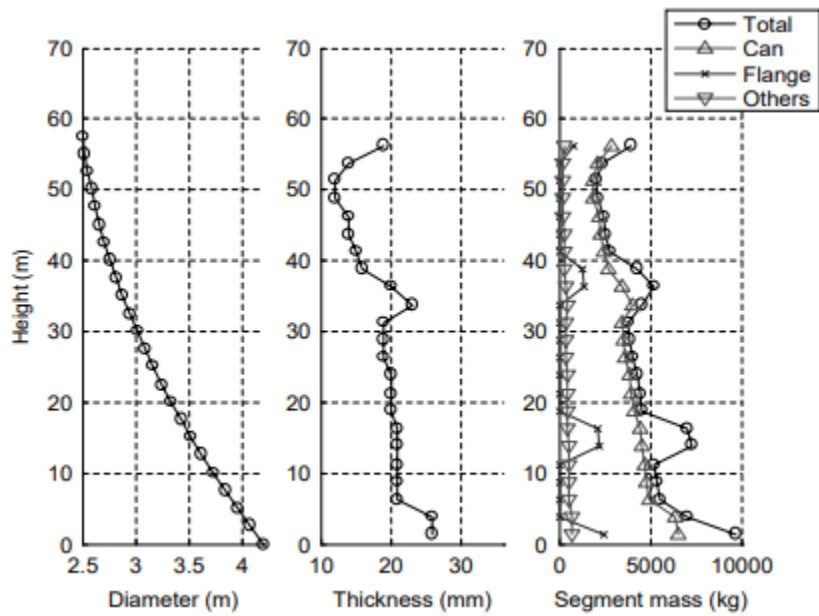


Figure 8: Tower G Results

A-Mazer

For this optimization problem, it is much like the traveling salesman in that the best route is to be determined. The difference being, there are only two location points. More thought is needed for this experiment on the part of the genetic algorithm, in the sense that there are walls in place that act as barriers to be eliminated in one spot. Meaning, one wall needs to be removed to make the maze functionable and the best option needs to be determined by the genetic algorithm.

Mazes are generated with random entry and exit locations for each new maze. The Genetic Algorithm must analyze the possible routes and find the shortest path.

Like in the previous problems, the solution goes through selection, crossover, and mutation processes. The GA also has the specific, tailored, fitness equation that is applied. Genetic algorithms can accommodate different types of problems.

An example of a finished maze completed with genetic algorithms is shown below in Figure 8.

While the genetic algorithm does not always find the best solution, it finds better paths compared to other tried applications.

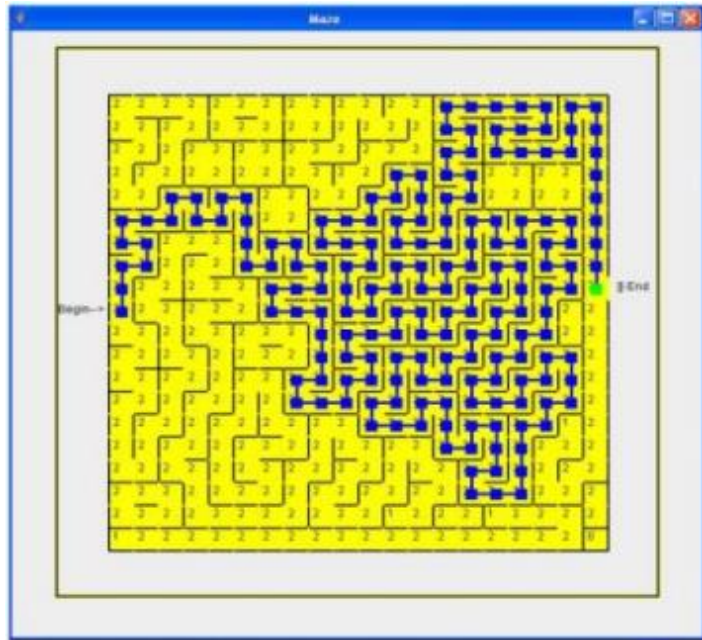


Figure 10: Maze Solution Example

Conclusions

While genetic algorithms can often be used to optimize at least one element of a problem, they have various strengths and limitations. Genetic algorithms have an evolutionary design that makes optimization problems a good match, there are some optimization problems that are better suited for genetic algorithms. Other “decision making” algorithms might be better suited for the other optimization problems. Genetic algorithms show promising results when facing these optimization problems, but some are more complex than others. Genetic algorithms are better suited for optimization problems that are harder to determine using traditional analytic techniques.

References

- Choubey, Nitin. (2012). A-Mazer with Genetic Algorithm. *International Journal of Computer Applications* (0975 – 8887). 58. 48-54. 10.5120/9378-3886.
- Hazim, Ziyad & Al-Jabbar, Abid. (2020). Using Genetic Algorithm to Solve Travelling Salesman Optimization Problem Based on Google Map Coordinates for Duhok City Areas.
- Mjahed, Mostafa. (2013). PID Controller Design using Genetic Algorithm technique.
- PETKOVIC, D., & RADOVANOVIC, M. (2013). Using Genetic Algorithms for Optimization of Turning Machining Process. *Journal of Engineering Studies & Research*, 19(1), 47–55.
- Shigeo Yoshida. (2006). Wind Turbine Tower Optimization Method Using a Genetic Algorithm. *Wind Engineering*, 30(6), 453.