



Team Carnation:  
Power Generation

Joseph Grosso      Kyle Singer      Andrew Fryer  
                         Andrew Farley

18th January 2020

# Contents

1	Introduction . . . . .	1
2	Considerations . . . . .	1
2.1	Energy Cost . . . . .	1
2.2	CO <sub>2</sub> Production . . . . .	2
2.3	Reliability . . . . .	2
3	Solution . . . . .	2
3.1	Algorithm . . . . .	2
3.2	Implementation . . . . .	3
4	Conclusion . . . . .	3

## 1 Introduction

Our challenge is to create an algorithm that acts as a single power grid for the entire province of Ontario. The program will address the energy needs of the province, taking into account the supply and demand of power from various sources. Our system addresses both the needs of the consumer (minimizing black/brownouts, minimizing price) and prioritizes green energy sources to limit environmental impact of the energy system.

## 2 Considerations

Our group took into account many issues to create our algorithm, which will be outlined below.

### 2.1 Energy Cost

Our algorithm had to minimize the cost of the energy production, to ensure that the Hydro System would not be losing money.

---

## 2.2 CO<sub>2</sub> Production

The algorithm also had to take into account the price of using sources of carbon emissions, contributing to the global warming crisis.

## 2.3 Reliability

Our customers need to be able to rely on the power grid. Our algorithm has to ensure that customers will not experience blackouts, brownouts, or power cuts of any kind.

# 3 Solution

## 3.1 Algorithm

The heart of the algorithm is a priority queue, ranking the energy sources by their benefits according to our considerations. Using these, we were able to create score functions for each power source providing us with our priority queue which we could draw power in order from.

The score functions looked like this:

$$\text{Green Score} = \begin{cases} 1 & \text{if energy source} \in \text{Solar, Wind, Hydro, Biofuel} \\ 0.8 & \text{otherwise} \end{cases}$$

$$\text{CO}_2 \text{ Score} = 1 - \left( \frac{\text{CO}_2 \text{Emissions}(\text{source})}{B \times \text{MAX}(\text{CO}_2 \text{Emissions})} \right)$$

$$\text{Cost Score} = 1 - \left( \frac{\text{cost}(s)}{C \times \text{MAX}(\text{cost})} \right)$$

$$\text{Source Score} = \text{Green Score} + \text{CO}_2 \text{ Score} + \text{Cost Score}.$$

Because all power created by the Nuclear source had to be used, in our algorithm we used all Nuclear power first and then split up the remaining power demand according to our priority. To decide how we should change the amount of Nuclear power generated on each iteration, we used the following equation:

$$\text{Movement\_Direction} = \frac{t_{n+1}}{t_n} - 1$$

$$\text{Nuc. Source Change} = \text{clamp}(\text{Movement\_Direction} \times \text{Nuc. Source} + \text{Nuc. Source}, -0.01, 0.01).$$

where  $t_n$  is the total energy usage as of one week ago, movement direction tells us the direction that the nuclear power generation will be moving in, and the clamp function caps the movement of the nuclear source at 1%.

---

### 3.2 Implementation

We implemented our algorithm in Python, using the libraries Pandas and Numpy. We also used Matplotlib to do visualizations of our data output.

## 4 Conclusion

We implemented our code and found it to be successful. We were able to generate an algorithm which balances the CO<sub>2</sub> emissions of the energy generation with selling enough power to be profitable in every hour (approx. \$410,000 per hour). While our CO<sub>2</sub> emissions were higher than expected we should be able to improve the algorithm in the future.

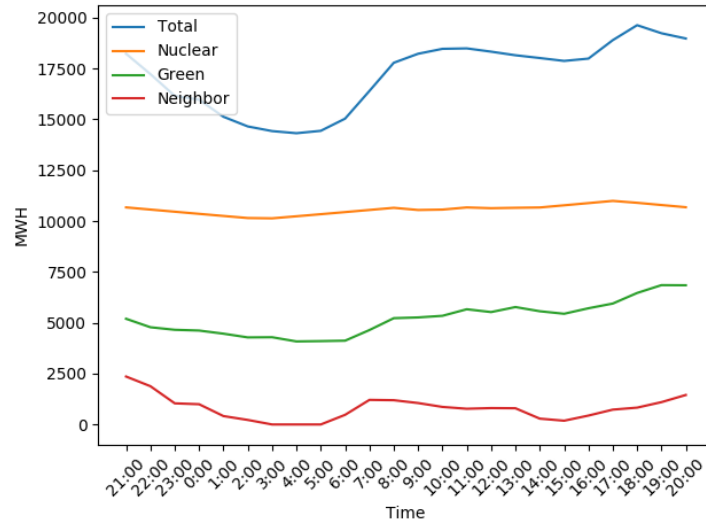


Figure 1: Time vs. MWH generated by each energy source

