# Hybrid Energy Sizing using Genetic Algorithm

Data Science for Mechanical Systems

Ansh Sandhu (acs2321), Harshit Kumar (hk3080)

hk3080/data-science-final-project (github.com)

## Table of Contents

# 1. Introduction

## a. The rationale behind the problem statement

The motivation behind this project is bringing more people within the umbrella of clean, reliable electricity which more than a billion people still lack [1]. We believe that social and economic development in the underdeveloped parts of the world, parts where people lack access to reliable electricity, would be a crucial battle in the war against Climate change. Our argument for the same, is the dynamics of population growth as the living standards increase.

Population growth has been seen as the major culprit behind global warming. More people need more resources. Economic growth however has been shown to stabilize population growth in the long run. In 1945, Princeton demographer Frank Notestein outlined a three-stage demographic model to illustrate the dynamics of population growth with economic development. It states that in pre-modern societies there is little or no population growth. In stage two, as living standards rise and health care improves, death rates begin to decline. With birth rates remaining high while death rates are declining, population growth accelerates, typically reaching close to 3 percent a year. As living standards continue to improve, and particularly as women are educated, the birth rate also begins to decline. Eventually the birth rate drops to the level of the death rate. This is stage three of the demographic transition, where births and deaths are in balance and population is again stable [2].

Thus, we believe that providing reliable energy might be a prerequisite for the economic growth of underdeveloped parts of the world, and hence an antidote to the rising population, and by extension, to global warming in the long term. Renewable energy might be especially lucrative for such communities as they are essentially a blank canvas for the technology without legacy power plants holding back their integration.

## b. Problem Statement

**Problem:** The greatest challenge to the widespread integration of renewables into the grids is their intermittency. The energy sources - wind and solar for this project - only produce energy when the sun shines, or the wind blows. Hence without cost-effective energy storage, their energy is at the mercy of weather conditions.

**Hypothesis:** We believe that diversifying the renewable technology portfolio might help in reducing the overall fluctuations in electricity produced. For example, Solar energy is generally stronger during the summer months in general and in noon in particular. Wind generally shows a complementary behavior: being stronger in the winter months and at night. Smart integration of both might be a problem worth considering.

Hence, using a bunch of different solar panels and wind turbines, each with their very own power curve, should minimize the localized intermittency. We can further reduce the fluctuations by diversifying the weather data itself: installing them at many different locations, reducing the total intermittency, and hence reducing the capacity of storage needed. We analyze this hypothesis by the problem statement below.

Combining these renewable energy sources into a hybrid system with energy storage can provide a more economic, environment friendly and reliable supply of electricity in all load demand conditions compared to single-use of such systems in the rural areas. One of the most important issues in this type of hybrid system is to optimally size the hybrid system components as sufficient enough to meet all load requirements with possible minimum investment and operating costs. For this project, we only consider capital cost in the analysis. For our analysis we are considering NYC electricity consumption given the ease of data collection.

**Problem Statement:** Given the target consumptions of the New York State and the weather data for its constituent cities, recommend the optimal renewable energy mixture.

## 2. Literature Review

The lifetime cost typically consists of two other components additional to the operational cost. These components include the cost of capital and maintenance cost, together referred as “fixed cost”. In calculation of the lifetime cost, changes within the price due to time should also be taken into consideration. Thus, the optimal hybrid system configuration seeks a mixture of generator types and sizes that yields lowest lifetime cost and/or emission. Among all possible hybrid system configurations that are optimally dispatched, the configuration with the least “Net Present Value (NPV)” is called as the “optimal configuration” or the “optimal design” [3,4].

Among them, the most famous sizing programs for hybrid systems is HOMER developed by National Renewable Energy Lab- oratory (NREL), United States [5]. HOMER has been wildly used in previous renewable energy system case studies that took place in the literature. Both grid-parallel and stand-alone systems had been investigated.

GA is a method of optimization which is based on the genetic process of biological organisms [6,7]. By mimicking this process, GA has capability to provide solutions to complex real-world problems. The concept of GA was firstly proposed by Holland [8] and then was widely used in many types of applications.

The input data of Genetic Algorithm- based methodology could be the meteorological conditions and the unit prices of the projected hybrid system components including installation and maintenance costs. The best advantage of GA to be used in hybrid system sizing is that it can easily jump out of a local minimum and it is quite efficient in finding the global optimum. GA has been widely used in several cases and information regarding its use can be found in many published articles dealing with the hybrid system sizing. Among them, Koutroulis et al. [8,9], Yang et al. [10,11] and Bilal et al. [12] utilized GA for sizing of a stand-alone hybrid PV–wind system. Lagorse et al. [13] applied GA to economically design a multisource hybrid unit composed of PV, wind and fuel cell. A more detailed system consisting of PV, wind, fuel cell, microturbine and battery was optimally sized by Kalantar et al. [14] using this algorithm.

# 3. Data and Methods:

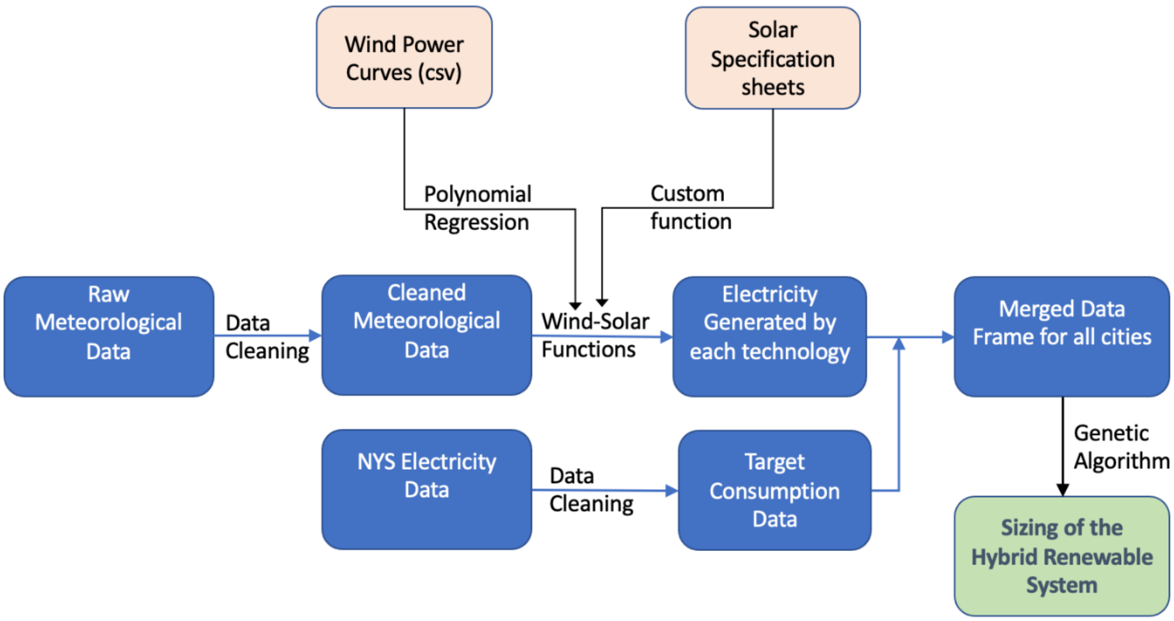



**Figure 3.1. Workflow of the Project**

## a. Data:

We collected and aggregated meteorological data (temperature, solar radiation, wind speeds) of five cities: Albany, Binghamton, Buffalo, Rochester and New York city [15]. NYS ISO maintains an extensive database including the hourly electricity consumption of the state [16]. For the genetic algorithm to identify a good fit, we must diversify the renewable energy portfolio. Thus, we used wind turbines from 20 kW-class to MW class to cover the spectrum. We selected 6 wind turbines namely- AWT-26, AWP 200/23, D2CF 200, AN Bonus 150/30, Aria 20 and WWD-1 D64.

The electricity produced by the solar PV is a function of both Solar Radiation and Temperature. The specifications of these panels – Cell temperature, temperature coefficients of voltage and current – decide how their electricity production changes. Since the electricity produced by PV can be algebraically added (in the absence of shading, resistances, etc), we aimed to diversify these specifications. Hence, we chose to diversify these specifications while looking for possible solar panels.

## b. Converting weather data into electricity produced:

**Wind:** Raw wind data can be converted into electricity produced by the wind turbine using power curves (plots that show energy corresponding to the wind speed at the turbine's hub). We converted the wind data into electricity data in following two steps:

1. Wind speed correction: The wind data obtained was measured at the height of 1m. To get the speed at the rotor hub, we used the wind profile power law. The logarithmic profile (or log law) assumes that the wind speed is proportional to the logarithm of the height above ground. The following equation therefore gives the ratio of the wind speed at hub height to the wind speed at anemometer height []:

$$\frac{U_{Hub}}{U_{ground}} = \left(\frac{Z_{hub}}{Z_{ground}}\right)^{\alpha}$$

2. Obtaining a continuous power curve: Wind turbines only operate in a window of the cut-in and cut-off speeds, outside which they produce no electricity. The electricity produced in the working range are available as dynamic plots on the manufacturer websites. We extracted discrete values from these plots [18] (energy corresponding to a wind speed) and used polynomial regression was used to fit the points and get the best possible power curve that we could. We used a very high degree

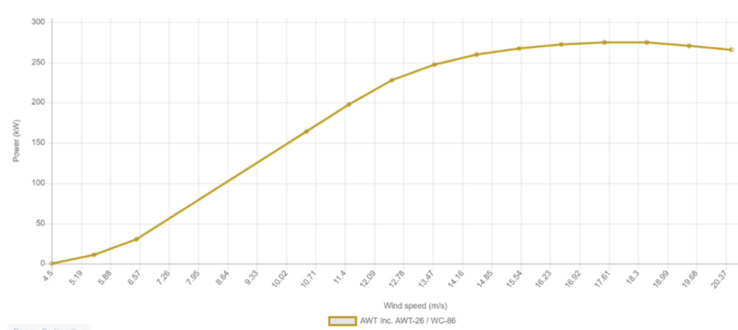



**Fig 3.2: Power curve example of an AWT-26 Wind Turbine:** We took discrete values from this graph and then generated a power curve using Polynomial regression.

of freedom (10 in our case) which was reasonable since we are not using the obtained fit to predict values outside the range. The accuracy within the range is our only concern.

3. Wind speed to electricity: Wind turbines behave differently beyond their working range. The values were assigned values according to their specification sheets. The values inside the range are obtained by predicting based on the fit.

**Solar:** We wrote a custom function to convert the raw solar data into electricity produced. In the analysis we have assumed ideal case scenarios of zero resistance and no shading. We also restricted the analysis to temperature coefficient of power produced ($Pmax$) instead of accounting for Open circuit voltage ($Voc$) and short circuit current ($Isc$) independently. The solar specification sheets give the power produced by the panel at Normal Operating Cell Temperature (NOCT). We used the following equations to get electricity:

$$T_{cell} = T_{ambient} + \frac{NOCT - 20}{800} \times Solar\ Radiation\ ({}^{W}/{}_{m^2})$$

$$P_{actual} = P_{\max @\, NOCT}\,(1 - (T_{cell} - NOCT) \times Temp\ Coefficient\ (P_{max}))$$

The functions written for both solar and wind were individually applied on the weather data to get the electricity produced by the given technology in a particular city.

## c. Normalizing the electricity production:

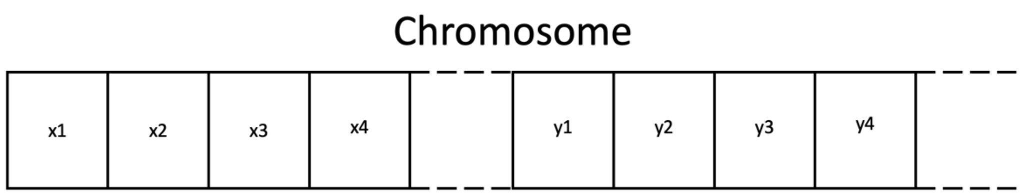
We decided to normalize our data by using the capital cost of each technology. For simplicity, we didn’t consider economies of scale, and assumed a fixed value per kW of capacity installed ($/kV). For wind we used $1.75 Million/MW for MW-class projects and $6000/kW for kW-class projects. To normalize solar, we assumed discrete solar farms of 5 kW at $10952 per farm.

## d. Optimizing the sizing of the hybrid system using the Genetic Algorithm:

Genetic Algorithms (GA) mimic the evolutionary process of natural selection. Just like nature selects the genes which are the fittest in that particular environment, the algorithm would select the fittest solution from a set of given solutions based on the fitness function defined.

**Step 1. Identifying and defining the solution space as a chromosome:**

Each solution (chromosome) has a size of 34 genes. Each gene represents the size of the technology it corresponds to.




$x_i$ represents the number of $i^{th}$ wind turbines installed, $y_i$ represents the number of $i^{th}$ solar panels installed.

Hence, the problem can be mathematically represented as:

$$Y = x_1 W_1 + x_2 W_2 + \cdots + y_1 S_1 + y_2 S_2 + \cdots + y_n S_n$$

Where, Y is the data frame with hourly electricity consumption of NYC; x and y represent the number of units for a particular renewable technology; W and S represent the data frames with the electricity produced by wind and solar technologies.

For the GA to converge to the right solution in a reasonable time, it is necessary to have a good idea of the range of each solution. We defined this range by dividing a representative value of the target consumption, by representative minimum and maximum values for each technology. This gave us a range of possible values. Through trial and error, we found that GA gives better results when the maximum value was given a conservative value and allowed to increase through mutations.

**Step 2: Defining a Fitness Function:**

We defined error as the difference between target electricity consumption and the electricity produced by the hybrid system. Since our goal is to provide reliable electricity every hour, both underproduction and overproduction of electricity are detrimental. Hence, we aimed to minimize the absolute value of the hourly error. Hence the fitness function was defined as:

$$error = |\, Electricity\ consumption - \ Elecricity\ produced\ by\ the\ hybrid\ system)\, |$$

$$Fitness = 1/error$$

**Step 3: Selecting the parents for mating:**

The algorithm then selects the given number of parents (4-8 in our case) based on their fitness values. These parents then mate with the rest of the solutions in the next step.

**Step 4: Crossover:**

Crossover is the process of creating new off springs, by creating combinations of parent's genes. The parents - chromosomes that are the fittest - would impart their genes onto future generations who might be fitter than their parents. The original chromosomes of the parents are also kept in the future generations. We define a division point between 2 and 4 across which the genes are shared. A division point of 2 would correspond to the genes switching half-way; 3 would mean genes switching over a third of the way, and so on.

**Step 5: Mutation:**

The Genetic Algorithm often gets stuck onto local maxima. It is especially difficult when we deal with columns that show correlations with one another (See results for correlations). Mutation is a way to get out of these local solutions. The mutation function allows some genes to randomly change with a “mutation probability”, which is a variable. The algorithm then randomly selects rows and columns to be mutated, and a random number to be added to that column.

We found that for sizing problems like ours, where the solution can only be positive, it was important to avoid negative mutations. The negative coefficients provide an excellent fit **Figure 3.3**, and the GA recognizes that very quickly.

Through experimentation we found that the conventional Genetic Algorithm locks on to a few negative coefficients pretty quickly to give a good, but useless fit. Even if we start with big and positive values, the mutations quickly realize the benefit of going negative and locks onto them.

To avoid this, we picked lower values for all coefficients and tweaked the mutation function to only 9allow gradual and positive increments.

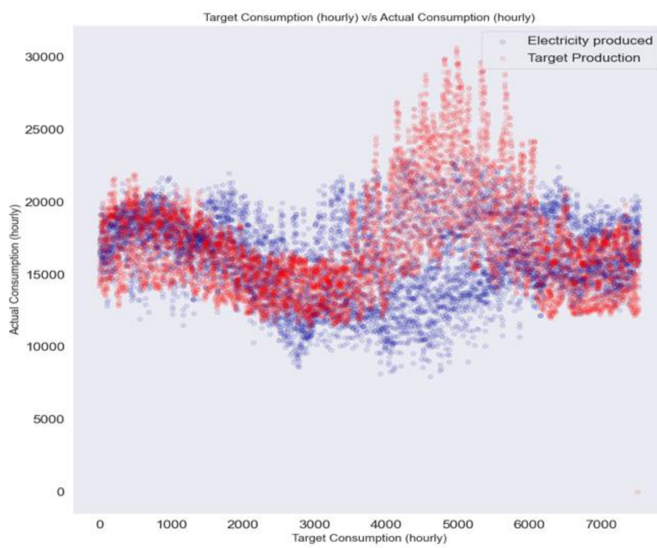



**Figure 3.3. Fit when negative coefficients are allowed:** The GA quickly locks on to such solutions when allowed to mutate organically

These 5 steps are carried out over a defined number of generations or until an optimal solution is reached. Given our restriction of computing power, we used a larger population (100) and relatively fewer generations (also 100) to reach a solution. This, along with the narrow solution space, allowed the GA to quickly find the best chromosomes and only rely on mutations for fine tuning. We found it to be a good tradeoff between accuracy and speed.

## e. Scope of improvement in our Genetic Algorithm:

a. The current mutation function adds a random value between 0 to 10 to a random column. The performance can be made by mutating the genes according to the solution space for that particular column.

b. Since we haven't considered energy storage in our model, the system algorithm might be biased towards instant gratification. The lack of a "buffer" in the energy storage looks for hourly, absolute error. Since the solar system sits idle for an average of 12+ hours in a day, the absence of this buffer might hit it especially hard, showing up as an aversion to solar energy in our model. A possible test for this problem has been discussed under "further work".

c. The crossover process might be made better by allowing similar genes to crossover: allowing the genes that correspond to wind technology to crossover with other wind technology genes.

# 4. Results:

## a. Considering all the cities:

The Genetic Algorithm quickly converges to a reasonable solution within a few generations. The mutations then allow it to fine-tune the sizes in small increments. The **Figure 4.1** shows the fitness as a function of the number of generations. The final portfolio and the corresponding energy production are shown in the **Figure 4.2**.

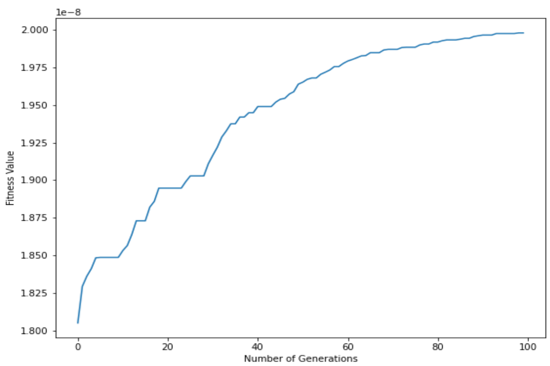



**Figure 4.1: Fitness value as a function of number of generations:**
The benefit of mutations is apparent in the few horizontal lines when the solution converges to a local max, but mutations allow the algorithm to explore the solution space

| | |
|---|---|
| AWT26_Power(albany) | 10 |
| AWP_200_Power(albany) | 159 |
| D2CF_200_Power(albany) | 47 |
| AN_Bonus_150/30_Power(albany) | 32 |
| Aria_20_Power(albany) | 1068 |
| WWD-1_D64_Power(albany) | 18 |
| Solar_AllMax(albany) | 31 |
| Solar_TrinaSolar(albany) | 64 |
| Solar_Mitsubishi(albany) | 47 |
| Solar_FirstSolar(albany) | 62 |
| AWT26_Power(buffalo) | 66 |
| AWP_200_Power(buffalo) | 114 |
| D2CF_200_Power(buffalo) | 31 |
| AN_Bonus_150/30_Power(buffalo) | 70 |
| Aria_20_Power(buffalo) | 911 |
| WWD-1_D64_Power(buffalo) | 2 |

| | |
|---|---|
| AWT26_Power(binghamton) | 76 |
| AWP_200_Power(binghamton) | 56 |
| D2CF_200_Power(binghamton) | 91 |
| AN_Bonus_150/30_Power(binghamton) | 32 |
| Aria_20_Power(binghamton) | 1180 |
| WWD-1_D64_Power(binghamton) | 18 |
| AWT26_Power(rochester) | 124 |
| AWP_200_Power(rochester) | 58 |
| D2CF_200_Power(rochester) | 26 |
| AN_Bonus_150/30_Power(rochester) | 55 |
| Aria_20_Power(rochester) | 743 |
| WWD-1_D64_Power(rochester) | 9 |
| AWT26_Power(nyc) | 86 |
| AWP_200_Power(nyc) | 73 |
| D2CF_200_Power(nyc) | 32 |
| AN_Bonus_150/30_Power(nyc) | 70 |
| Aria_20_Power(nyc) | 604 |
| WWD-1_D64_Power(nyc) | 15 |

**Figure 4.2: Sizing of the hybrid system after 100 generations:**
The GA selects more units of wind power than its solar counterparts. It also picks a greater number of smaller 20 kW turbines, which might occur because we haven’t considered economies of scale in the MW turbines

The **Figure 4.3** shows the hourly electricity produced by our combination in the absence of any energy storage. Integrating energy storage and natural gas fired plants would allow us to manage the fluctuations and minimize the curtailing.

The Genetic Algorithm shows a preference for wind energy in NYC. That is intuitive because NYC has a longer winter than the summer: more potential for wind energy than solar energy. The summer demand could theoretically be met by importing solar energy from “sunnier states”. This helps make a case for a more flexible energy flow across the U.S. state lines.

The error as a function of seasons (day of the year) is shown in the **Figure 4.4**. The error is the highest in the months from June-September, when the wind is at its weakest.

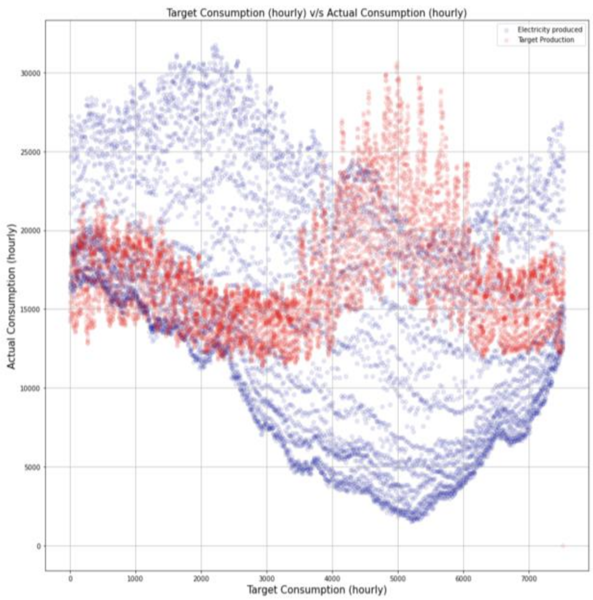



**Figure 4.3: Hourly Target Consumption (red) vs. Electricity Produced by the Hybrid System (blue):** The Algorithm recommends pushing wind in NYC to meet the energy demands. This is intuitive since NYC weather

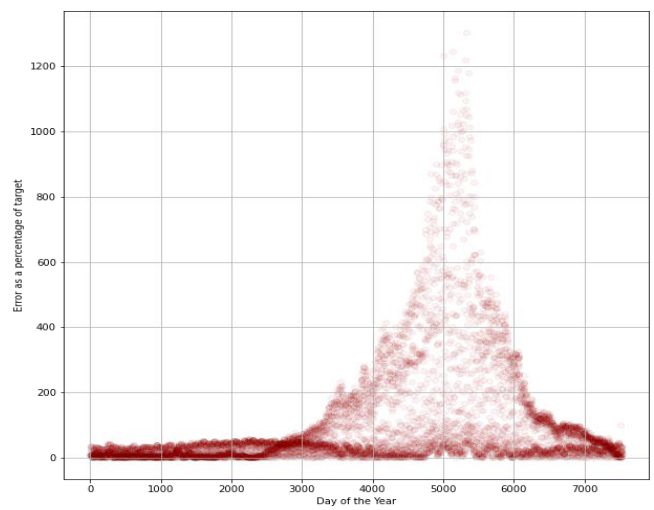



**Figure 4.4: Hourly absolute error (% of target consumption) in electricity production:** The error is centered around the months of June-September, when the wind is the weakest (observed from the electricity produced by each individual wind turbine)

To cover up for the error in the summer, we tried incentivizing the solar resources by giving them an edge in the mutation process, but the algorithm never picked up those solutions. This might be because increments in solar comes at the cost of less fitness in winter (given the under production in the winter months), hence the algorithm suggests against investing in solar power in NYC. Alternatively, this might be due to neglecting the economies of scale in our normalizing step (See further work).

## b. Prioritizing the cities based on correlation matrix:

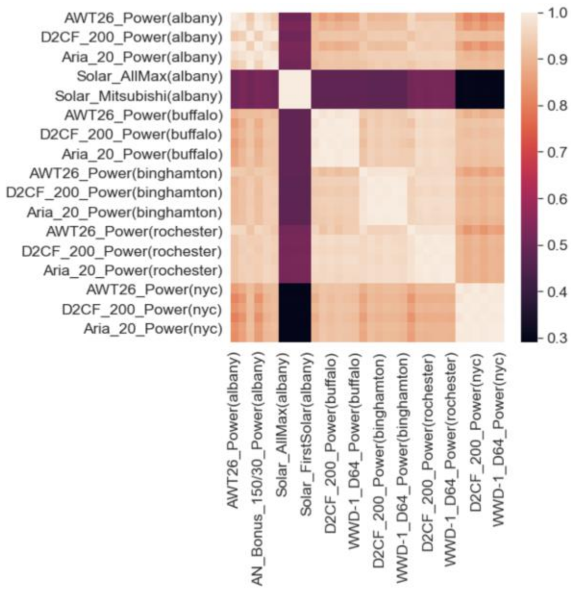




**Figure 4.5: Correlation in the electricity produced in different cities**

The heatmap shows the correlation of electricity produced in different cities. Cities with similar geographies - e.g., Rochester and buffalo - have high correlation. NYC has lower correlations with other cities possibly due to the proximity to the Atlantic Ocean. The analysis of two cases: all cities (high correlation) and only NYC and Albany (low correlation), has been presented in the Results. The heatmap also makes a case for greater flexibility: the farther and geographically distinct the mixture, lesser the correlation.

## c. Considering the cities with low correlation: Albany and NYC

The decrease in complexity of the problem allows the Genetic Algorithm to run the algorithm over a greater number of generations (**Figure 4.6**). Indeed, we get a slightly better fit given more running time to fine tune the fit. The "better fit" can be judged by comparing the values of the fitness in both graphs.

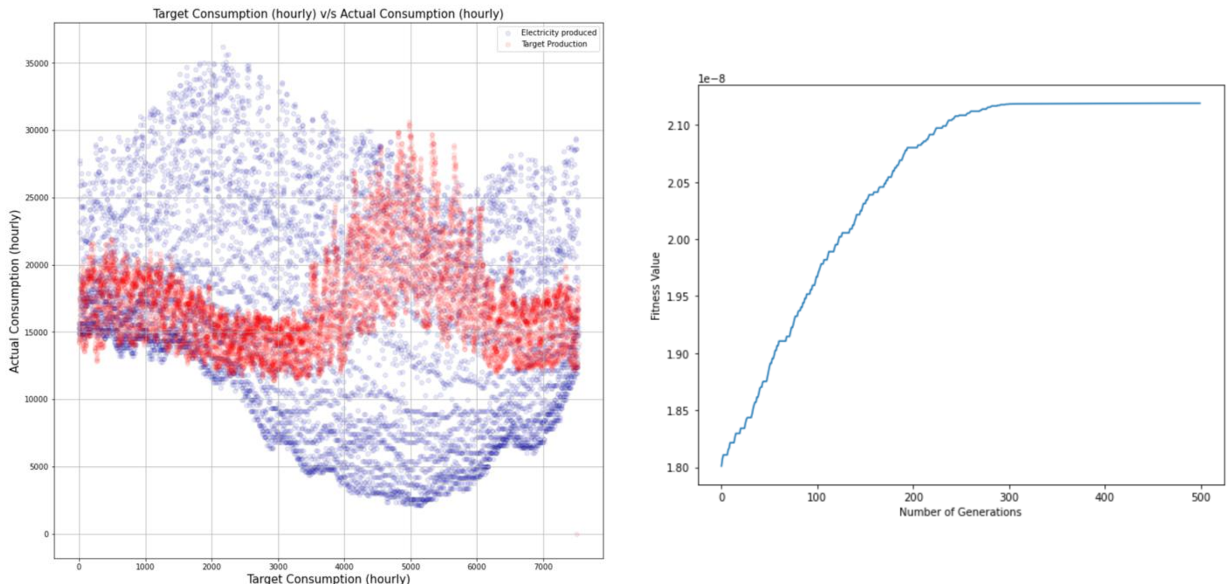



**Figure 4.6. Results for running the genetic algorithm:** Slightly better fit as we could perform more computations in the same amount of time

The fitness of the lower dimensional data (2.11920571e-08) is slightly higher than for the higher dimensional, correlated data (1.99854621e-08). Hence, it is better to remove the data that shows high correlation. However, this should not be taken as a case against diversifying the weather base, but against choosing cities with correlated data. This makes a case for diversifying the generation capacity discussed in the section below.

This distance would also bring in a need for a stronger, smarter transmission infrastructure. Not to mention the need for frequency regulation and ancillary grid services to make this integration possible. The cost-benefit analysis of such an endeavor is out of scope for this project.

# 5. Conclusion:

The genetic algorithm is an efficient way of obtaining the sizing of the hybrid renewable energy system. Using a primitive genetic algorithm, we could get a reasonably good fit of the target. The current algorithm showed a preference on investing in wind power for NYC. Removing the cities with less correlated weather data showed similar results, much faster.

Our results support diversifying the renewable technology as well as locations of installation to reduce the sizing of the energy storage required.

# 6. Further work:

1. The efficacy of our Genetic Algorithm can be tested by **running it on solar dominated regions** like California and New Delhi. Such regions have summer dominated months which can be quantified by an architectural concept: Hot degree Days (HDD) and Cooling Degree Days.

The regions dominated by Summer are typically driven by cooling requirements in the summer, which also peaks with solar electricity. Hence, our algorithm should converge to a solar energy dominated solution for such regions.

2. Our current study assumes fixed capital cost for solar and wind resources. It **doesn’t consider economies of scale** with wind and solar farms. Including those numbers to build a more dynamic and robust model would be interesting.

3. The current model **doesn’t include energy storage** to buffer the hourly energy production. Including the effect of energy storage within fitness function: allowing delayed gratification, would greatly benefit this approach. The delayed gratification might also benefit solar PV installation as they can produce excess energy during the day, which can be stored and used up later.

4. Considering **offshore weather data** in NYC’s possible wind portfolio.

# References:


1. F Birol, editor. World Energy Outlook 2010.. International Energy Agency, 2010

2. Frank Notestein, "Population—The Long View," in P. W. Schultz, ed., Food for the World (University of Chicago Press: 1945); Population Action International (PAI), "How Demographic Transition Reduces Countries' Vulnerability to Civil Conflict," fact sheet (Washington, DC: updated 17 December 2004).

3. Anglani N, Muliere G. Analyzing the impact of renewable energy technologies by means of optimal energy planning. In: 9th international conference on environment and electrical engineering (EEEIC). 2010. p. 1–5.

4. Mizani S, Yazdani A. Design and operation of a remote microgrid. In: 35th annual conference on industrial electronics (IECON'09). 2009. p. 4299–304

5. Razali NMM, Hashim AH. Backward reduction application for minimizing wind power scenarios in stochastic programming. The 4th interna- tional power engineering and optimization conf (PEOCO'2010) 2010: 430–4.

6. Goncalves JF, Mendes JJM, Resende MGC. A genetic algorithm for the resource constrained multi-project scheduling problem. European Journal of Opera- tional Research 2008;189(3):1171–90.

7. Chen SM, Huang CM. A new approach to generate weighted fuzzy rules using genetic algorithms for estimating null values. Expert Systems with Applica- tions 2008;35(3):905–17

8. Holland JH. Adaptation in natural and artificial systems. Massachusetts, Cam- bridge: MIT Press; 1975

9. Koutroulis E, Kolokotsa D, Potirakis A, Kalaitzakis K. Methodology for opti- mal sizing of stand-alone photovoltaic/wind-generator systems using genetic algorithms. Solar Energy 2006;80(9):1072–88

10. Koutroulis E, Kolokotsa D. Design optimization of desalination sys- tems power-supplied by PV and W/G energy sources. Desalination 2010;258(1–3):171–81

11. Yang H, Zhou W, Lu L, Fang Z. Optimal sizing method for stand-alone hybrid solar–wind system with LPSP technology by using genetic algorithm. Solar Energy 2008;82(4):354–67.

12. Yang H, Zhou W, Lou C. Optimal design and techno-economic analy- sis of a hybrid solar–wind power generation system. Applied Energy 2009;86(2):163–9.

13. Bilal BO, Sambou V, Ndiaye PA, Kébé CMF, Ndongo M. Optimal design of a hybrid solar–wind–battery system using the minimization of the annualized cost system and the minimization of the loss of power supply probability (LPSP). Renewable Energy 2010;35(10):2388–90.

14. Lagorse J, Paire D, Miraoui A. Hybrid stand-alone power supply using PEMFC, PV and battery – modelling and optimization. In: International conference on clean electrical power (ICCEP). 2009. p. 135–40.

15. https://search.dataone.org/view/p1008.ds2043

16. https://search.dataone.org/view/p1008.ds2043

17. https://www.homerenergy.com/products/pro/docs/latest/wind_resource_variation_with_height.html

18. https://en.wind-turbine-models.com/powercurves