Optimal Decarbonisation Software Framework User Manual

In colloboration with:

UNIVERSITY OF SURREY

University of Nottingham UK | CHINA | MALAYSIA

東京大学 THE UNIVERSITY OF TOKYO

MONASH University

DE LA SALLE UNIVERSITY MANILA

BRITISH COUNCIL

Purusothmn Nair A/L S Bhasker Nair

8/13/2021

Purpose of Document

This document serves as a user guide for the optimal decarbonisation software framework developed in this research project. The decision-making software framework is based on rigorous mathematical optimisation models, for planning the decarbonisation of ASEAN countries, in line with commitments made while signing the Paris Agreement. The planning framework relies on a combination of proven, mature technologies such as the Carbon Emission Pinch Analysis (CEPA), developed by members of the project team over the past 10 years, and novel mathematical optimisation-based tools that provide rigorous guarantees on the qualities of the solution, subject to planning constraints such as budget, social resistance to uptake, efficiencies of interventions, and implementation time.

The open-source software and planned impact activities ensure that the results and tools' impact is maximised to help both governmental and industrial policymakers in ASEAN countries to identify achievable emissions targets and the optimal paths to achieve them through a range of technologies, interventions, and budgetary and time constraints. The team will deliver significant outreach and engagement activities through multi-day workshops with project partners in emissions-intensive industries in Malaysia, as well as with government agencies to ensure that the software and solutions are data-driven, implementable and align with national strategies.

Further information regarding this research project may be obtained from the [project website]() and [GitHub](). Further questions about this research project may be directed to Dr Michael Short (m.short@surrey.ac.uk).

Acknowledgement

The team would like to offer their sincerest gratitude to the British Council for funding provided towards the COP26 Trilateral Research Initiative. The team would like to also take this opportunity to thank the project partners i.e., Malaysian Green Technology and Climate Change Centre, Aria Sustainability Ltd, Professor Raymond Tan and Dr Jully Tan for their continuous support and valuable inputs throughout this project. Sincere gratitude is equally extended to Pyomo, Octeract and GitHub for the use of their tools towards the successful execution of this research project.

Table of Contents

1. Introduction

Climate change is the gravest threat to humanity's long-term prosperity and the global cooperation to mitigate this threat is unprecedented. The **COP26 summit** taking place in November 2021 aims to accelerate the achievement of the Paris Agreement's goals. To achieve these ambitious emissions targets and keep global warming below 2°C, strategic planning methods for policymaking are essential. These should span entire nations' emissions contributions, across sectors, and should be able to plan for achievable implementation of emissions reduction technologies, negative emissions technologies (NETs), within budgetary, time, and social uptake constraints. Association of South-East Asian Nations (ASEAN) countries, as developing economies, have seen dramatic rises in CO_2 emissions over the past 20 years e.g., CO_2 per capita of Malaysia has risen from 5 t y^{-1} in year 2000 to 8 t y^{-1} in year 2018. Therefore, it is important to develop an energy planning tool that incorporates region-specific conditions.

This research project seeks to develop a decision-making software framework, based on rigorous mathematical optimisation models, for planning the decarbonisation of ASEAN countries, in line with commitments made while signing the Paris Agreement. The planning framework relies on a combination of proven, mature technologies such as the Carbon Emission Pinch Analysis, developed by members of the project team over the past 10 years. Additionally, this research project would also utilise novel mathematical optimisation-based tools that provide rigorous guarantees on the qualities of the solution, subject to planning constraints such as budget, social resistance to uptake, efficiencies of interventions and policies, uncertainty, technology readiness levels, and implementation time. The open-source software and planned impact activities ensure that the results and tools' impact is maximised to help both governmental and industrial policymakers in ASEAN countries to identify achievable emissions targets and the optimal paths to achieve them through a range of technologies, interventions, and budgetary and time constraints.

This **research project** brings together academic, industrial and government agencies for the development of the decision-making software framework. The research project team is as follows:

Principal Investigator

Dr Michael Short (University of Surrey, UK)

Co-Investigators

Professor Dominic Foo (University of Nottingham Malaysia)

Dr Yasunori Kikuchi (University of Tokyo)

Lead Developer

Purusothmn Nair (University of Nottingham Malaysia)

Project Partners

Malaysian Green Technology and Climate Change Centre

Dr Matteo Cossutta (Aria Sustainability Ltd)

Professor Raymond Tan (De La Salle University, Manila)

Dr Jully Tan (Monash University Malaysia)

2. Optimal Decarbonisation Software Framework

The optimal decarbonisation software developed in this research project is based on a multiperiod energy planning framework. Energy sources i.e., renewable energy, natural gas, oil, and coal each with different output ranges are utilised to satisfy the demand in a geographical region. The satisfaction of the CO_2 emission limit is achievable with the use of energy-producing NETs (EP-NETs), energy-consuming NETs (EC-NETs) and carbon capture and storage (CCS). The optimal decarbonisation software framework is built based on the potential deployment of various EP-NETs, EC-NETs and CCS technologies.

2.1. Energy Planning Parameters

The parameters of the optimal decarbonisation software framework are presented in Table 1. The values of each parameter should be declared in the Microsoft Excel file titled '**Energy_Planning_User_Interface**'. The Microsoft Excel file consists of three worksheet tabs e.g., '**PLANT_DATA**', '**EP_DATA**' and '**NET_DATA**'. The worksheet tab '**PLANT_DATA**' consists of the parameters associated with each plant e.g., carbon intensity, energy output bounds, CCS data etc. On the other hand, the worksheet tab '**EP_DATA**' is made of energy planning constraints e.g., budget, demand, and emission limit. Meanwhile, the worksheet tab '**NET_DATA**' consists of data related to the various NETs utilised in this research project.

Table 1: Energy Planning Parameters

Parameters	Definition
$C_{S,i}$	Carbon intensity of power plant *i*
$F_{i,LB}$	Lower bound of energy output by power plant *i*
$F_{i,UB}$	Upper bound of energy output by power plant *i*
RR_n	Removal ratio of CCS technology *n*
X_n	The parasitic power loss of CCS technology *n*
$Cost_i$	Cost of energy output by power plant *i*
$Cost_{i,n}$	Cost of energy output by power plant *i* with CCS technology *n*
D	The demand of a geographical region
L	CO_2 emission limit in a geographical region
$CIEC_q$	Carbon intensity of EC-NET technology *q*
$CIEP_p$	Carbon intensity of EP-NET technology *p*
$CTEC_q$	Cost of EC-NET technology *q*
$CTEP_p$	Cost of EP-NET technology *p*

$\boldsymbol{CIC}$	Carbon intensity of compensatory energy
$\boldsymbol{CTC}$	Cost of compensatory energy
$\boldsymbol{BD}$	Budget allocation

2.2. Energy Planning Variables

The variables in the optimal decarbonisation software framework are presented in Table 2. These variables should be defined before optimising the mathematical formulation.

Table 2: Energy Planning Variables

Variables	Definition
$\boldsymbol{C_{R,i,n}}$	Carbon intensity of power plant *i* with CCS technology *n*
$\boldsymbol{F_{S,i}}$	Energy output by power plant *i*
$\boldsymbol{F_{R,i,n}}$	The extent of CCS retrofit of power plant *i* with CCS technology *n*
$\boldsymbol{F_{R,i}}$	The total extent of CCS retrofit of power plant *i* with all CCS technologies
$\boldsymbol{F_{NR,i,n}}$	Net energy output by power plant *i* with CCS technology *n*
$\boldsymbol{F_{NR,i}}$	Total net energy output by power plant *i* with all CCS technologies
$\boldsymbol{F_{NS,i}}$	Net energy output by power plant *i* without CCS retrofit
$\boldsymbol{B_{i,n}}$	Binary variable for selection of power plant *i* with CCS technology *n*
$\boldsymbol{F_{EP,p}}$	Minimum deployment of EP-NET technology *p*
$\boldsymbol{F_{EC,q}}$	Minimum deployment of EC-NET technology *q*
$\boldsymbol{F_{C}}$	Minimum deployment of compensatory energy
$\boldsymbol{TE}$	Total CO_2 emissions at the end of energy planning
$\boldsymbol{TC}$	Total energy costs at the end of energy planning

2.3. Mathematical Formulation

This section presents the constraints involved in the mathematical formulation of the optimal decarbonisation framework. Firstly, the summation of the energy output from power plant *i* must be equivalent to the demand of a specified geographical region for each period *k*, as shown in Equation 1. Also, the energy output from power plant *i* should be in the range of lower and upper boundary of energy output for each period *k*, as demonstrated in Equation 2.

$$\left(\sum_{i} F_{S,i} = D\right)_k \qquad \forall k \qquad \text{Equation 1}$$

$$\left(F_{i,LB} \leq F_{S,i} \leq F_{i,UB}\right)_k \qquad \forall i\ \forall k \qquad \text{Equation 2}$$

Next, the carbon intensity of power plant *i* with CCS technology *n* for each period *k* is determined from Equation 3.

$$\left[C_{R,i,n} = \frac{C_{S,i} \times (1 - RR_n)}{1 - X_n}\right]_k \qquad \forall i\ \forall n\ \forall k \qquad \text{Equation 3}$$

The net energy output from power plant *i* with CCS technology *n* for each period *k* is calculated from Equation 4. Note that the reduced energy output from power plant *i* is due to the parasitic power losses during CCS. The total energy output from power plant *i* with all CCS technologies for each period *k* is calculated from Equation 5.

$$\left[F_{R,i,n} \times (1 - X_n) = F_{NR,i,n} \times B_{i,n}\right]_k \qquad \forall i\ \forall n\ \forall k \qquad \text{Equation 4}$$

$$\left(\sum_{n} F_{NR,i,n} = F_{NR,i}\right)_k \qquad \forall i\ \forall k \qquad \text{Equation 5}$$

The summation of the extent of CCS retrofit of power plant *i* with all CCS technologies is calculated from Equation 6. Also, the total extent of CCS retrofit of power plant *i* should not exceed the energy output from power plant *i* for each period *k*, as shown in Equation 7.

$$\left(\sum_{n} F_{R,i,n} = F_{R,i}\right)_k \qquad \forall i\ \forall k \qquad \text{Equation 6}$$

$$\left(F_{R,i} \leq F_{S,i}\right)_k \qquad \forall i\ \forall k \qquad \text{Equation 7}$$

The summation of the net energy output by power plant *i* without CCS retrofit and the extent of CCS retrofit of power plant *i* with CCS technology *n* should equate to the energy output from power plant *i* for each period *k*; shown in Equation 8.

$$\left[F_{NS,i} + \sum_{n}\left(F_{R,i,n} \times B_{i,n}\right) = F_{S,i}\right]_k \qquad \forall i\ \forall k \qquad \text{Equation 8}$$

The summation of the energy output from all energy sources e.g., compensatory energy, EP-NET etc. must be equivalent to the total demand of the energy system e.g., EC-NET etc. for each period *k*, as demonstrated in Equation 9. Equally, the total CO_2 load contribution from all energy sources is equivalent to the total CO_2 emissions at the end of energy planning for each period *k*, shown in Equation 10.

$$\left[\sum_{i}\sum_{n}\left(F_{NS,i} + \left(F_{NR,i,n}\, B_{i,n}\right)\right) + F_C + \sum_{p} F_{EP,p} = \sum_{q} F_{EC,q} + D\right]_k \qquad \forall k \qquad \text{Equation 9}$$

$$\left[\sum_i \sum_n \left(F_{NS,i} C_{S,i} + \left(F_{NR,i,n}\, C_{R,i,n}\right)\right) + F_C\, CIC + \sum_p F_{EP,p}\, CIEP_p + \sum_q F_{EC,q}\, CIEC_q = TE \right]_k \qquad \forall k \qquad \text{Equation 10}$$

Meanwhile, the total energy costs at the end of energy planning are calculated from Equation 11.

$$\left[\sum_i \sum_n \left(F_{NS,i} Cost_i + \left(F_{NR,i,n}\, Cost_{i,n}\right)\right) + F_C\, CTC + \sum_p F_{EP,p}\, CTEP_p + \sum_q F_{EC,q}\, CTEC_q = TC \right]_k \qquad \forall k \qquad \text{Equation 11}$$

Due to economic concerns, it is necessary to ensure that the extent of CCS retrofit on power plant *i* at a later period is at least equal to that in its previous period, as shown in Equation 12. This ensures that a decision taken to CCS retrofit power plant *i* in period *k* would not be overturned in the subsequent periods.

$$\left(F_{R,i}\right)_{k+1} \geq \left(F_{R,i}\right)_k \qquad k = 1,2,\dots,n-1 \qquad \text{Equation 12}$$

The constraints regarding the total CO_2 emissions and total energy costs are presented in Equation 13 and Equation 14 respectively.

$$(TE = L)_k \qquad \forall k \qquad \text{Equation 13}$$

$$(TC \leq BD)_k \qquad \forall k \qquad \text{Equation 14}$$

The next section discusses the objective function of the optimal decarbonisation software framework.

2.4. Objective Function

The mathematical formulation in Section 2.3 may be optimised according to either Equation 15 or Equation 16, depending on the user preference. For Equation 15, the total energy cost is minimised subject to constraints in Equation 1 till Equation 13. In other words, the minimisation of the total energy costs would ensure that the emission limit in a geographical region for each period *k* is satisfied. Meanwhile, for Equation 16, the total emission is minimised subject to the constraints in Equation 1 till Equation 12, and Equation 14. In other words, the minimisation of the total emission is conducted subject to the budgetary constraint for each period *k*. Therefore, the emission limit in a geographical for each period *k* may or may not be satisfied. Before optimising the mathematical formulation, a user should specify the objective function by selecting a drop-down box on cell '**B33**' in the worksheet tab '**PLANT_DATA**'. There are two optimisation choices provided i.e., '**min_budget**' (Equation 15) and '**min_emission**' (Equation 16).

$$\min TC \quad \text{Equation 15}$$

$$\min TE \quad \text{Equation 16}$$

Since $B_{i,n}$ is a binary variable for the deployment of CCS technology n in power plant i, the resulting mathematical formulation is a mixed-integer non-linear programming (MINLP) model. The next section describes the step-by-step procedure to download the relevant files for optimal decarbonisation from GitHub.

2.5. GitHub Files Download

All files relevant to the optimal decarbonisation framework should be downloaded from GitHub to a user's operating machine before performing any optimisation task. The step-by-step procedure for the download of the relevant files from GitHub is as follows:

i. A user should access the BCCOP26 Trilateral Project in GitHub from https://github.com/mchlshort/BCCOP26TrilateralProject.

ii. A user had arrived at the right landing page if the top right of the GitHub page is labelled as '**mchlshort/BCCOP26TrilateralProject**'

iii. A user should click the green icon '**Code**', followed by '**Download ZIP**' to download all files relevant to the optimal decarbonisation framework, as shown in Figure 1.

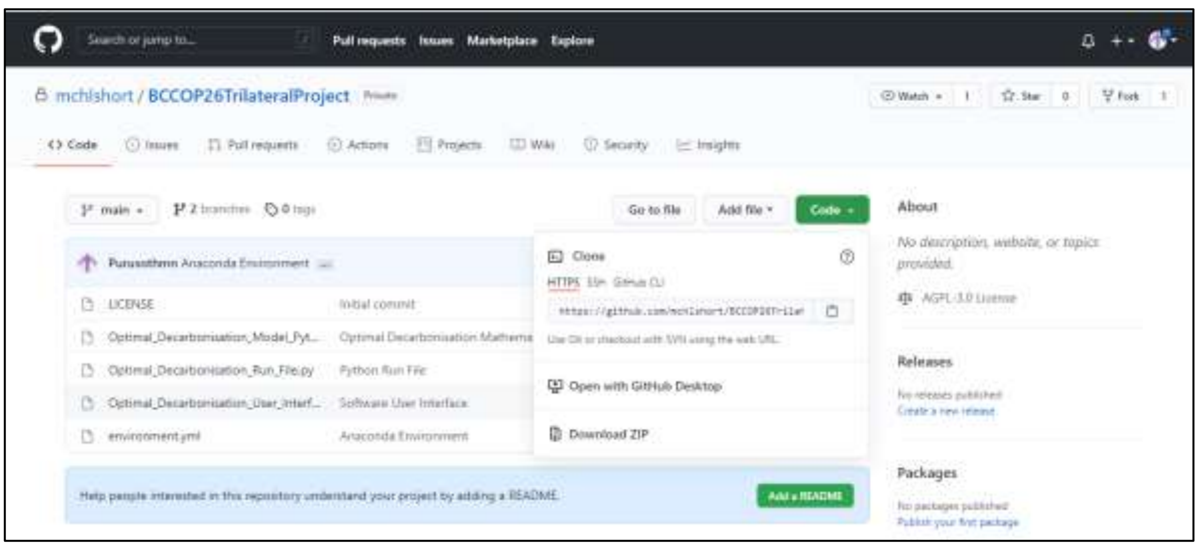

Figure 1: GitHub Landing Page

iv. The ZIP folder downloaded from GitHub would contain four files, described as follows:

- **Optimal_Decarbonisation_User_Interface** – This file contains the user interface for the optimal decarbonisation software framework. A user should input the plant and energy planning data in this file before performing the optimisation in Python.

- **Optimal_Decarbonisation_Model_Python** – This file contains the complete mathematical formulation for the optimal decarbonisation software framework.

- **Optimal_Decarbonisation_Run_File** – This file contains the run code for the optimal decarbonisation framework in Python

- **environment.yml** – This file contains the environment to be imported to Anaconda

v. A user should ensure that the files are saved in the same directory that would be used later for the Octeract Engine solver files.

The MINLP model is solved using **Pyomo, a Python-based open-source optimisation modelling language**. The following section describes the installation procedure of Anaconda on a user's operating machine.

2.6. Anaconda Installation

Anaconda is a free and open-source distribution for Python programming language to be used for scientific computing. Anaconda contains multiple data-science packages that may be used in Windows, Linus and macOS. Therefore, the first step would be for a user to install Anaconda on one's operating machine. The step-by-step procedure for Anaconda installation is as follows:

vi. A user should install Anaconda from https://www.anaconda.com/distribution/.

vii. Once download, a user should launch the Anaconda application and view the home page as shown in Figure 2.

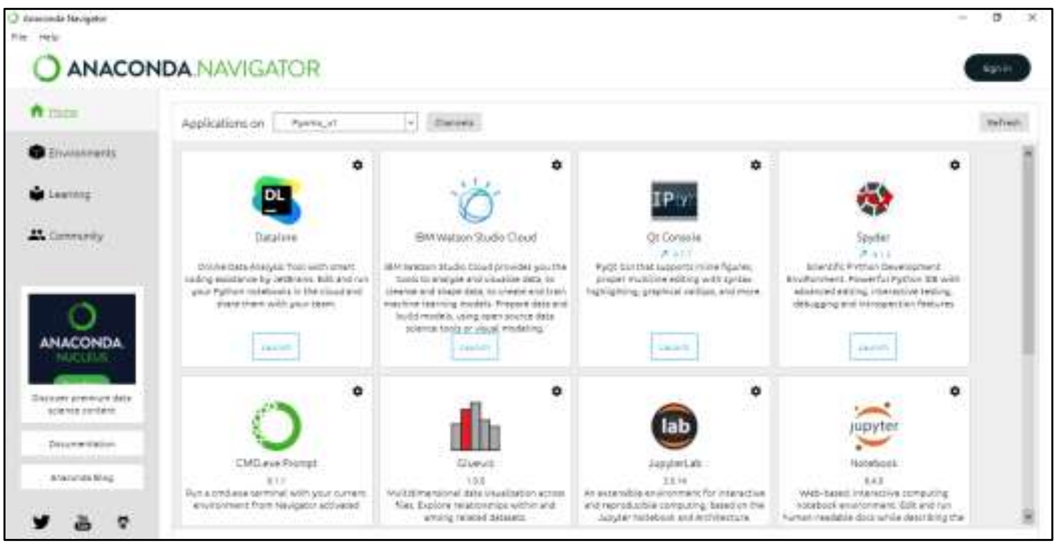

Figure 2: Anaconda Home Page

viii. Under the top-left of the home page, a user should click the '**Environment**' tab.

ix. At the bottom of the '**Environment**' tab, a user would see four options provided to create a new environment in Anaconda. (see Figure 3)

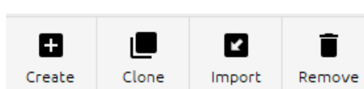

Figure 3: Options for new Anaconda Environment

x. A user should click on '**Import**' and arrive at the '**Import new environment**' pop-up window visualised in Figure 4.

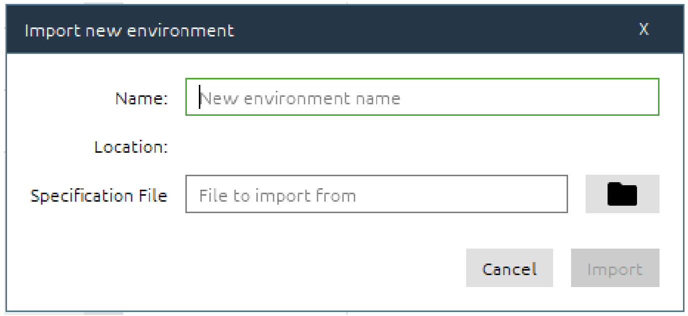

Figure 4: Import new environment

xi. The name of the new environment may be customised according to a user's preference e.g., *Energy Planning*, *Modelling* etc.

xii. Under the '**Specification File**' box, a user should import the environment file titled '**environment.yml**'.

xiii. Once the environment file is loaded, a user should click import. The import process should take several minutes to be completed.

xiv. Once the import process is completed, a user should be able to see the name of the new environment below the default (base) Anaconda environment.

xv. A user should select the new environment. If this new environment was not previously selected, this process of activating the new environment should take approximately 30 seconds.

xvi. Next, a user should click the right arrow of the new environment and select '**Open Terminal**', as shown in Figure 5.

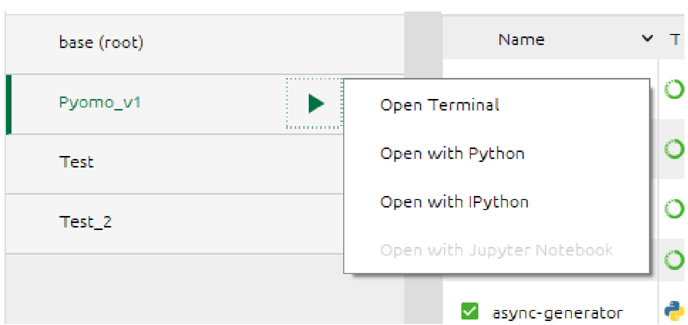

Figure 5: Options for the new environment

xvii. A pop-up window would appear as shown in Figure 6. A user should ensure that the terminal is loaded for the correct environment. The name of the new environment should appear in a bracket before the directory specification (see Figure 6).

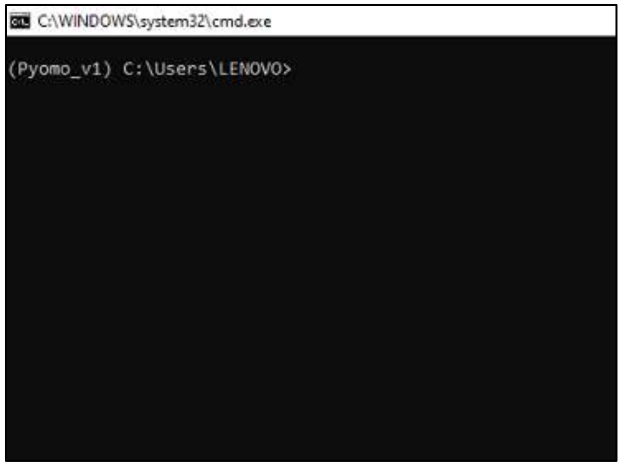

Figure 6: The terminal pop-up window

xviii. Under the terminal, a user should type '**Spyder**' and click enter, as shown in Figure 7.

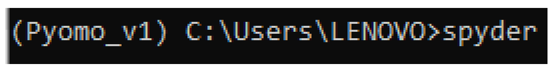

Figure 7: Launching Spyder

xix. The Spyder application would be launched, and a user would be able to view the Spyder console, as shown in Figure 8.

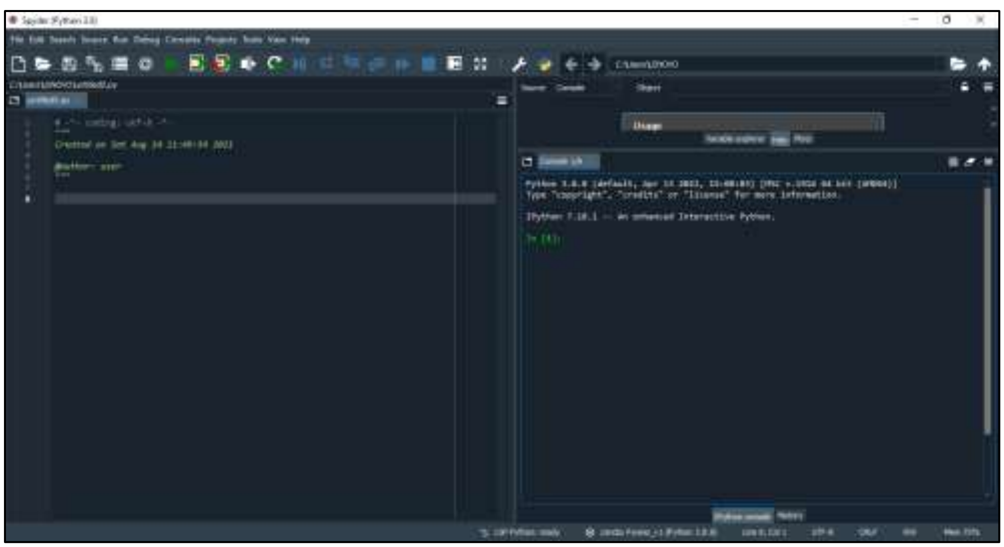

Figure 8: Spyder console

At this step, a user should be able to use Spyder to solve an optimisation problem. Since the mathematical formulation for this research project is an MINLP model, a user should install an appropriate optimisation solver i.e., Octeract Engine. The following section describes the installation procedure of Octeract Engine for the Windows operating system.

2.7. Octeract Engine Installation for Windows Operating System

Octeract Engine is the optimisation solver for the MINLP mathematical formulation in this research project. Since Octeract Engine is not readily available in Python, a user should install the Octeract Engine solver beforehand. The step-by-step procedure for the Octeract Engine installation for the Windows operating system is as follows:

ii. A user should install the Octeract Engine solver from https://octeract.com/#download
iii. A user should also complete the registration process at https://octeract.com/register to obtain the authentication token to be used during the engine setup.
iv. During installation, the user would be prompted to choose an authentication token, as shown in Figure 9.
v. The user should choose the second option '**Use an existing authentication token**'.
vi. The user would then be prompted to select a file containing the authentication token downloaded from the website upon completion of registration.

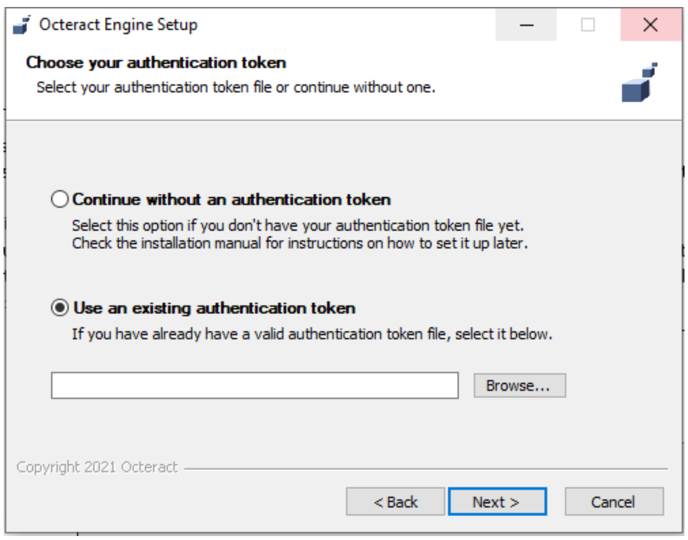

Figure 9: Octeract Engine Authentication Token

vii. Note, the Octeract Engine solver should be installed in the same directory (folder) that was used to save the optimisation files downloaded from GitHub.
viii. For a smoother user experience, a user should set the path directory of the environment variables.
ix. A user should search for '**environment variables**' and arrive at the page shown in Figure 10.

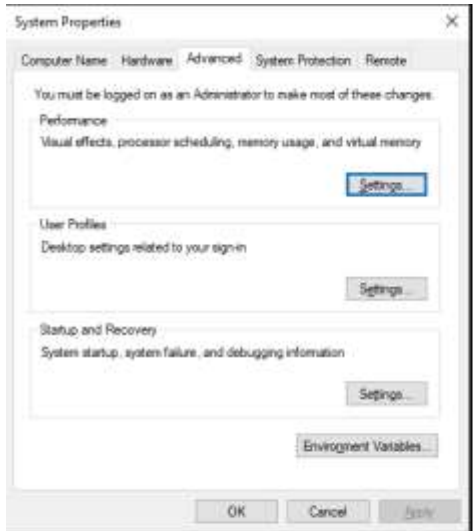

Figure 10: System Properties

x. Next, the user should click the ‘**Environment Variables**’.

xi. Under the ‘user variables for user’ box, a user should select ‘**Path**’ and click ‘**Edit**’, as shown in Figure 11.

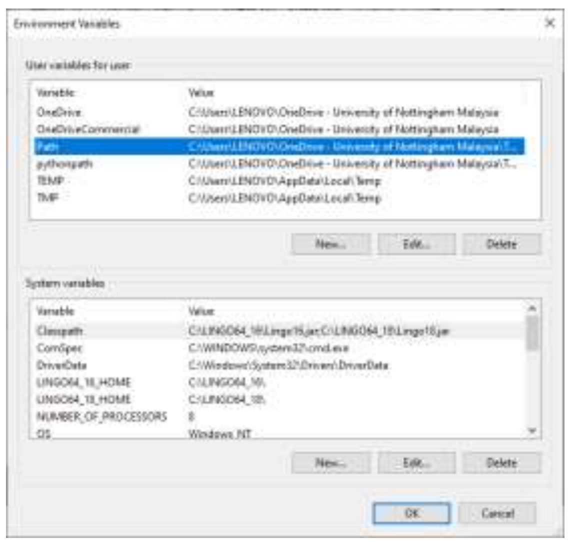

Figure 11: Environment Variables

xii. Then, a user should click ‘**New**’ to add the directory that the necessary files for the optimal decarbonisation software framework e.g., *Energy_Planning_Model_Python*, *Energy_Planning_Run_File*, Octeract Engine etc. (see Figure 12)

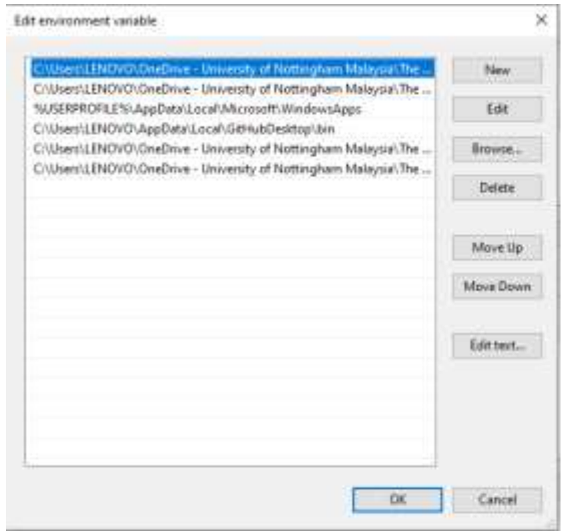

Figure 12: Edit Environment Variables

The following section describes the installation procedure of Octeract Engine for the Mac operating system.

2.8. Octeract Engine Installation for Mac Operating System

Presently, the Octeract Engine is not supported for the Mac operating system. A Mac user could use a Windows or Linus virtual machine to install the Octeract Engine solver (refer to Section 2.7 for the installation procedure). The next section describes the step-by-step procedure of optimising the mathematical formulation in Section 2.3, based on the objective function in Section 2.4.

2.9. Mathematical Optimisation

The optimisation of the decarbonisation framework would be conducted via Spyder. The step-by-step procedure for the mathematical optimisation is as follows:

i. In Spyder, a user should open the file titled '**Optimal_Decarbonisation_Run_File**' by selecting '**Open**' under the '**File**' tab.
ii. Once the '**Optimal_Decarbonisation_Run_File**' file is opened, a user should edit the pathname. An example of a pathname would be '**C:/Users/Lenovo/Energy Optimisation**' if the relevant files (Octeract Engine solver, GitHub files etc.) are saved in a folder called '**Energy Optimisation** under the 'C' drive.
iii. Another option to obtain the file directory would be to right-click the folder that contains the relevant files and click '**Properties**'.
iv. The '**Location**' represents the file directory, as shown in Figure 13.

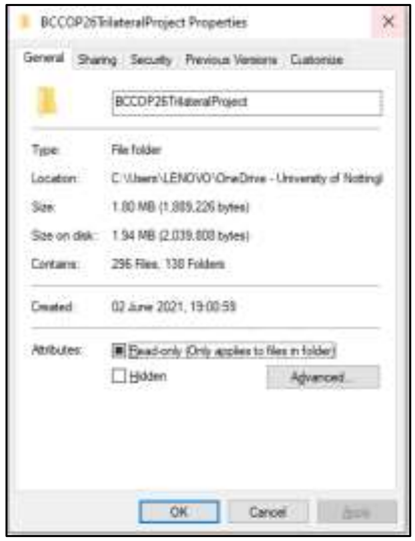

Figure 13: File Directory

v. Before optimising the decarbonisation framework, a user should input the plant and energy planning data in the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**'. The user should also specify the objective function. Kindly refer to Section 2.4 for more details regarding the objective function.

vi. Once the user had inputted the plant and energy planning data, a user should close the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**'.

vii. A user should then click F5 to run the file in Spyder. The optimisation procedure is complete once the solver status is displayed, as shown in Figure 14.

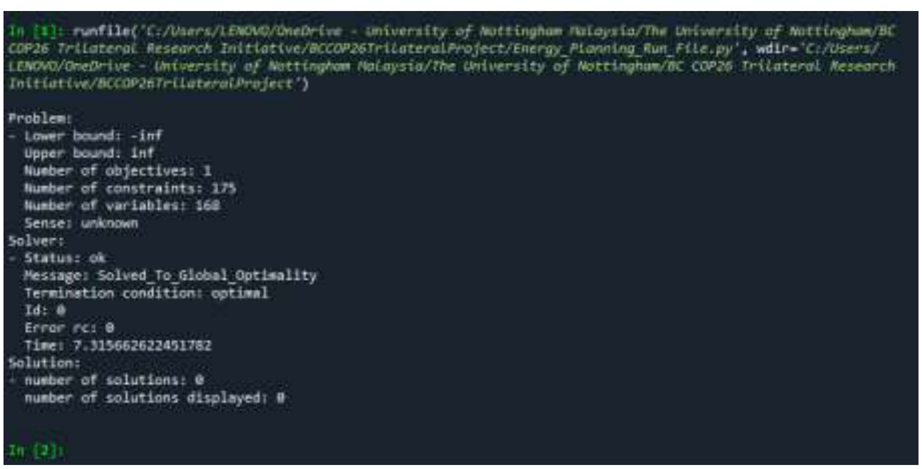

Figure 14: Solver Status

viii. A user should then re-open the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**' to view the detailed results.

ix. A user would observe additional worksheet tabs beyond the '**NET_DATA**' tab consisting of the optimisation results for each period.

The next section presents Case Study 1 to demonstrate the optimisation of the decarbonisation framework via Python.

3. Case Study 1

The application of the optimal decarbonisation software framework is demonstrated with a hypothetical case study i.e., Case Study 1. In this case study, seven power plants with a mix of renewable energy sources, natural gas, oil, and coal generate electricity for a geographical region. The energy planning is conducted across three periods, each with a specified demand, emission limit and budget allocation. The CO_2 load reduction is achievable with the potential deployment of three types of EP-NETs and EC-NETs, alongside two choices of CCS technology. The EP-NETs and EC-NETs vary in terms of their carbon intensities and cost for each period. Meanwhile, the CCS technology differs in terms of its removal ratio, parasitic power losses and cost. EP-NETs may consist of bioenergy with CCS (BECCS) and biochar. Meanwhile, EC-NETs may consist of direct air capture (DAC), enhanced weathering etc. Also, there is a choice of incorporating additional renewable energy (compensatory energy) for satisfying the CO_2 emission limit. The superstructure optimisation of the MINLP model would provide an overview regarding the optimal deployment of CCS, EP-NETs, EC-NETs, and compensatory energy for each period.

3.1. Power Plant Data

The power plant data for Case Study 1 are presented in Table 3.

Table 3: Power Plant Data for Case Study 1

	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6	Plant 7	Plant 8
Fuel type	Renewable	Natural Gas	Natural Gas	Natural Gas	Oil	Coal	Coal	Coal
Energy Output Lower Bound / TWh y^{-1}	10	0	0	0	0	0	0	0
Energy Output Upper Bound / TWh y^{-1}	40	40	40	40	10	20	20	20
Carbon Intensity / Mt TWh^{-1}	0	0.5	0.5	0.5	0.8	1.0	1.0	1.0
Removal Ratio of CCS technology 1	0	0.85	0.85	0.85	0.85	0.85	0.85	0.85
Power loss of CCS technology 1	0	0.15	0.15	0.15	0.15	0.15	0.15	0.15
Removal Ratio of CCS technology 2	0	0.65	0.65	0.65	0.65	0.65	0.65	0.65
Power loss of CCS technology 2	0	0.25	0.25	0.25	0.25	0.25	0.25	0.25

Fuel Cost / mil RM y^{-1}	155	104	104	104	202	51	51	51
Cost of CCS technology 1 / mil RM y^{-1}	0	141	141	141	280	82	82	82
Cost of CCS technology 2 / mil RM y^{-1}	0	120	120	120	240	66	66	66

Based on Table 3, only power plant 1 utilises renewable energy for electricity generation. Natural gas and coal are deployed in three power plants each while oil is solely utilised in power plant 5. Note that the lower bound for energy output concerning power plant 1 (renewable energy) is 10 TWh y^{-1}. This would ensure the deployment of renewable energy sources for each period during energy planning. Among all energy sources, coal is the cheapest but with the highest carbon intensity. On the other hand, CCS technology 1 is more expensive than CCS technology 2. However, CCS technology 1 has a greater removal ratio with a lower parasitic power loss in comparison to CCS technology 2. Therefore, there are pros and cons of deploying CCS technologies 1 and 2. A snapshot of the plant data in the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**' is shown in Figure 15. Note that the plant data should be inputted in the worksheet tab titled 'PLANT_DATA'.

TABLE 1: PLANT ENERGY PLANNING PARAMETERS

	Plant_1	Plant_2	Plant_3	Plant_4	Plant_5	Plant_6	Plant_7	Plant_8
Fuel	REN	NG	NG	NG	OIL	COAL	COAL	COAL
LB	10	0	0	0	0	0	0	0
UB	40	40	40	40	10	20	20	20
CI	0	0.5	0.5	0.5	0.8	1	1	1
RR_1	0	0.85	0.85	0.85	0.85	0.85	0.85	0.85
X_1	0	0.15	0.15	0.15	0.15	0.15	0.15	0.15
RR_2	0	0.65	0.65	0.65	0.65	0.65	0.65	0.65
X_2	0	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Cost	155	104	104	104	202	51	51	51
Cost_CCS_1	0	141	141	141	280	82	82	82
Cost_CCS_2	0	120	120	120	240	66	66	66

Figure 15: Snapshot of Plant Data in Microsoft Excel

3.2. NETs Data

The NETs data for Case Study 1 are presented in Table 4.

Table 4: NETs Data for Case Study 1

	Carbon Intensity / Mt TWh^{-1}	Cost / mil RM y^{-1}
EP-NETs 1	-0.8	175
EP-NETs 2	-0.6	165
EP-NETs 3	-0.4	150
EC-NETs 1	-0.6	200

EC-NETs 2	-0.4	150
EC-NETs 3	-0.2	100

Table 4 presents three technology choices of EP-NETs and EC-NETs. Each NETs vary in terms of carbon intensity and cost. Note that NETs with the highest carbon intensity is the most expensive.

3.3. Energy Planning Data

The energy planning data for Case Study 1 are presented in Table 5.

Table 5: Energy Planning Data for Case Study 1

Period	Demand / TWh y^{-1}	Emission limit / Mt y^{-1}	Budget/mil RM y^{-1}	Compensatory energy carbon intensity / Mt TWh^{-1}	Compensatory energy cost/mil RM y^{-1}
1	60	15	6,000	0	155
2	75	8	8,000	0	155
3	90	3	10,000	0	155

Based on Table 5, the optimal decarbonisation is conducted across three periods. Between successive periods, the energy demand rises by 15 TWh y^{-1} with a reduction of the CO_2 emission limit. The budget allocations also increase between successive periods. Due to the energy requirements of EC-NETs, compensatory energy with zero carbon intensity and a cost of 155 mil RM y^{-1} is available for deployment. The next section discusses the optimisation results for Case Study 1

3.4. Optimisation Results

The optimisation of Case Study 1 is conducted for both objective functions i.e., minimum budget and minimum emissions. For the minimum budget objective function, the emission limit for each period is satisfied. The total cost for periods **1, 2 and 3 are 5717, 7649 and 9,885 mil RM y^{-1} respectively**. For period 1, only CCS technology 1 is deployed for coal-based power plants 6 and 8. The deployment of CCS technology 2 only takes place in period 3 for natural-gas power plant 4.

Meanwhile, for the minimum emissions objective function, the optimisation is conducted subject to the budget constraints for each period. Therefore, the CO_2 emission for each period may or may not be violated. The total CO_2 emissions **in periods 1,2 and 3 are 8.9, 1.58 and 1.23 Mt y^{-1} respectively**. In other words, the total CO_2 emissions for all periods are lower than the CO_2 emission limit. Figure 16 presents a sample snapshot for the minimum budget optimisation results for period 1.

	Fuel	Energy	CI	CCS_1 CI	CCS_2 CI	CCS_1 Selection	CCS_2 Selection	CCS_1 Ret	CCS_2 Ret	Net Energy wo CCS	Net Energy w CCS	Net Energy	Carbon Load	Cost
Plant_1	REN	10	0	0	0	0	0	0	0	10	0	10	0	
Plant_2	NG	0	0.5	0.088	0.233	0	0	0	0	0	0	0	0	
Plant_3	NG	0	0.5	0.088	0.233	0	0	0	0	0	0	0	0	
Plant_4	NG	0	0.5	0.088	0.233	0	0	0	0	0	0	0	0	
Plant_5	OIL	0	0.8	0.141	0.373	0	0	0	0	0	0	0	0	
Plant_6	COAL	20	1	0.176	0.467	1	0	20	0	0	17	17	3	
Plant_7	COAL	10	1	0.176	0.467	0	0	0	0	10	0	10	10	
Plant_8	COAL	20	1	0.176	0.467	1	0	17.23	0	2.77	14.65	17.41	5.35	
EP_NET_1	EP_NET_1											0	0	
EP_NET_2	EP_NET_2											5.59	-3.35	
EP_NET_3	EP_NET_3											0	0	
EC_NET_1	EC_NET_1											0	0	
EC_NET_2	EC_NET_2											0	0	
EC_NET_3	EC_NET_3											0	0	
COMP	COMP											0	0	
TOTAL													15	5717.82

Figure 16: Minimum Budget Optimisation Results for Period 1

Based on Figure 16, note that the total CO_2 load and cost for each period may be observed at the bottom right of the optimisation results in Microsoft Excel. A similar format may be observed for the remaining worksheet tabs containing the optimisation results.