Optimal Decarbonisation Software Framework User Manual

In colloboration with:

Purusothmn Nair A/L S Bhasker Nair

10/7/2021

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 from the University of Surrey (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 (MGTC), 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 novel mathematical optimisation-based tool is expected to 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.

Table 1: Energy Planning Parameters

Parameters	Definition
$\boldsymbol{CS_i}$	Carbon intensity of power plant *i*
$\boldsymbol{F_{i,LB}}$	Lower bound of energy output by power plant *i*
$\boldsymbol{F_{i,UB}}$	Upper bound of energy output by power plant *i*
$\boldsymbol{RR_n}$	Removal ratio of CCS technology *n*
$\boldsymbol{X_n}$	The parasitic power loss of CCS technology *n*
$\boldsymbol{CT_{i,k}}$	Cost of energy output by power plant *i* in period *k*
$\boldsymbol{CTR_{i,k,n}}$	Cost of energy output by power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{CFX_{i,k,n}}$	Fixed cost of power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{D_k}$	The demand for an energy planning system in period *k*
$\boldsymbol{L_k}$	CO_2 emission limit an energy planning system in period *k*
$\boldsymbol{CIEC_{k,q}}$	Carbon intensity of EC-NETs technology *q* in period *k*
$\boldsymbol{CIEP_{k,p}}$	Carbon intensity of EP-NETs technology *p* in period *k*
$\boldsymbol{CTEC_{k,q}}$	Cost of EC-NETs technology *q* in period *k*
$\boldsymbol{CTEP_{k,p}}$	Cost of EP-NETs technology *p* in period *k*
$\boldsymbol{CIC_{k,r}}$	Carbon intensity of compensatory energy *r* in period *k*

$\boldsymbol{CTC_{k,r}}$	Cost of compensatory energy *r* in period *k*
$\boldsymbol{CIAS_{i,k,w}}$	Carbon intensity of alternative fuel *w* for solid fuelled power plant *i* in period *k*
$\boldsymbol{CTAS_{i,k,w}}$	Cost of alternative fuel *w* for solid fuelled power plant *i* in period *k*
$\boldsymbol{CIAG_{i,k,v}}$	Carbon intensity of alternative fuel *v* for gas-fuelled power plant *i* in period *k*
$\boldsymbol{CTAG_{i,k,v}}$	Cost of alternative fuel *v* for gas-fuelled power plant *i* in period *k*
$\boldsymbol{BD_{k}}$	Budget allocation in period *k*

The values of each parameter should be declared in the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**'. The Microsoft Excel file consists of six worksheet tabs e.g., '**PLANT_DATA**', '**EP_DATA**', '**ENERGY_DATA**', '**ALT_SOLID**', '**ALT_GAS**', '**CCS_DATA**', '**CI_NET_DATA**' and '**COST_NET_DATA**'. The data to be included in each worksheet tab is explained in Table 2.

Table 2: Data Inclusion for each Worksheet Tab in the User Interface

Worksheet Tab	Data Inclusion
PLANT_DATA	Fuel type, carbon intensity and energy output bounds for each plant
EP_DATA	Budget, demand, emission limit and choice of compensatory energy for a specified period
ENERGY_DATA	Fuel cost for a specified period
ALT_SOLID	Alternative fuel for solid fuelled based plants e.g., biomass
ALT_GAS	Alternative fuel for gas-fuelled based plants e.g., biomethane
CCS_DATA	Removal ratio, parasitic power loss and costs of CCS technologies for a specified period
CI_NET_DATA	Carbon intensities of NETs for a specified period
COST_NET_DATA	Costs of NETs for a specified period

2.2. Energy Planning Variables

The variables in the optimal decarbonisation software framework are presented in Table 3. These variables should be defined before optimising the mathematical formulation.

Table 3: Energy Planning Variables

Variables	Definition
$\boldsymbol{CR_{i,n}}$	Carbon intensity of power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{FS_{i,k}}$	Energy output by power plant *i* in period *k*
$\boldsymbol{FR_{i,k,n}}$	The extent of CCS retrofit of power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{FR_{i,k}}$	The total extent of CCS retrofit of power plant *i* with all CCS technologies in period *k*
$\boldsymbol{FNR_{i,k,n}}$	Net energy output by power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{FNS_{i,k}}$	Net energy output by power plant *i* without CCS retrofit in period *k*
$\boldsymbol{FAS_{i,k,w}}$	Minimum deployment of alternative fuel *w* for solid fuelled power plant *i* in period *k*
$\boldsymbol{FAG_{i,k,v}}$	Minimum deployment of alternative fuel *v* for gas-fuelled power plant *i* in period *k*
$\boldsymbol{B_{i,k,n}}$	Binary variable for selection of power plant *i* with CCS technology *n* in period *k*
$\boldsymbol{FEP_{k,p}}$	Minimum deployment of EP-NETs technology *p* in period *k*
$\boldsymbol{FEC_{k,q}}$	Minimum deployment of EC-NETs technology *q* in period *k*
$\boldsymbol{FC_{k,r}}$	Minimum deployment of compensatory energy *r* in period *k*
$\boldsymbol{TE_{k}}$	Total CO_2 emissions at the end of energy planning in period *k*
$\boldsymbol{TC_{k}}$	Total energy costs at the end of energy planning in period *k*

2.3. Mathematical Formulation

This section presents the constraints involved in the mathematical formulation of the optimal decarbonisation framework. For period k, the summation of the energy output from power plants $i \in I$ must be equivalent to the demand of a specified geographical region (D_k) as shown in

Equation 1. Also, the energy output from power plant *i* in period *k* ($FS_{i,k}$) should be in the range of lower ($F_{i,LB}$) and upper bound of energy output ($F_{i,UB}$).

$$\sum_{i} FS_{i,k} = D_k \qquad \forall k \tag{Equation 1}$$

Next, the carbon intensity of power plant *i* with CCS technology *n* in period *k* ($CR_{i,n}$) is determined from Equation 2.

$$CR_{i,n} = \frac{CS_i \times (1 - RR_n)}{1 - X_n} \qquad \forall i \; \forall n \tag{Equation 2}$$

The net energy output from power plant *i* with CCS technology *n* in period *k* ($FNR_{i,k,n}$) is calculated from Equation 3. Note that the reduced energy output from power plant *i* is due to the parasitic power losses during CCS. Also, $FNR_{i,k,n}$ should not exceed its upper bound of energy output in period *k*, as shown in Equation 4.

$$FR_{i,k,n} \times (1 - X_n) = FNR_{i,k,n} \qquad \forall i \; \forall k \; \forall n \tag{Equation 3}$$

$$FR_{i,k,n} \leq F_{i,UB} \times B_{i,k,n} \qquad \forall i \; \forall k \; \forall n \tag{Equation 4}$$

The summation of the extent of CCS retrofit of power plant *i* with all CCS technologies in period *k* ($FR_{i,k}$) is calculated from Equation 5. Also, the total extent of CCS retrofit of power plant *i* should not exceed the energy output from power plant *i* in period *k,* as shown in Equation 6.

$$\textstyle\sum_n FR_{i,k,n} = FR_{i,k} \qquad \forall i \; \forall k \tag{Equation 5}$$

$$FR_{i,k} \leq FS_{i,k} \qquad \forall i \; \forall k \tag{Equation 6}$$

For a given period *k*, the summation of the net energy output by power plant *i* without CCS retrofit ($FNS_{i,k}$) and the extent of CCS retrofit of power plant *i* with CCS technology *n* ($FR_{i,k,n}$) should equate to the energy output from power plant *i*; shown in Equation 8. This equation is only applicable for power plants utilising mixed or liquid-based fuels. Equation 8 and Equation 9 are applied to incorporate the deployment of an alternative low CO_2 intensity solid and gas-based fuel respectively.

$$FNS_{i,k} + \sum_{n} FR_{i,k,n} = FS_{i,k} \quad \forall k; i = mixed\ or\ liquid\ fuel\ plant \tag{Equation 7}$$

$$FNS_{i,k} + \sum_{n} FR_{i,k,n} + \sum_{w} FAS_{i,k,w} = FS_{i,k} \quad \forall k\ ; i = solid\ fuel\ plant \tag{Equation 8}$$

$$FNS_{i,k} + \sum_{n} FR_{i,k,n} + \sum_{v} FAG_{i,k,v} = FS_{i,k} \quad \forall k\ ;\ i = gas\ fuel\ plant \tag{Equation 9}$$

For a given period k, the summation of the energy output from all energy sources e.g., compensatory energy ($FC_{k,r}$), EP-NETs ($FEP_{k,p}$) etc. must fulfil the total demand of the energy system; the latter includes the total power requirement (D_k) and that required by EC-NETs ($FEC_{k,q}$) etc. as demonstrated in Equation 10. 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 period k (TE_k), shown in Equation 11.

$$\sum_i \sum_n \left(FNS_{i,k} + FNR_{i,k,n}\right) + \sum_i^{Solid\,fuel} \sum_w FAS_{i,k,w} + \sum_i^{Gas\,fuel} \sum_v FAG_{i,k,v} + \sum_r FC_{k,r} + \sum_p FEP_{k,p} = \sum_q FEC_{k,q} + D_k \quad \forall k \qquad \text{Equation 10}$$

$$\sum_i \sum_n \left(FNS_{i,k} CS_i + \left(FNR_{i,k,n}\, CR_{i,n}\right)\right) + \sum_i^{Solid\,fuel} \sum_w FAS_{i,k,w} CIAS_{i,k,w} + \sum_i^{Gas\,fuel} \sum_v FAG_{i,k,v}\, CIAG_{i,k,v} + \sum_r FC_{k,r}\, CIC_{k,r} + \sum_p FEP_{k,p}\, CIEP_{k,p} + \sum_q FEC_{k,q}\, CIEC_{k,q} = TE_k \quad \forall k \qquad \text{Equation 11}$$

Meanwhile, the total energy costs at the end of energy planning in period k (TC_k) are calculated from Equation 12.

$$\sum_i \sum_n \left(FNS_{i,k} CT_{i,k} + \left(FNR_{i,k,n}\, CTR_{i,k,n}\right) + \left(CFX_{i,k,n} B_{i,k,n}\right)\right) + \sum_i^{Solid\,fuel} \sum_w FAS_{i,k,w} CTAS_{i,k,w} + \sum_i^{Gas\,fuel} \sum_v FAG_{i,k,v}\, CTAG_{i,k,v} + \sum_r FC_{k,r}\, CTC_{k,r} + \sum_p FEP_{k,p}\, CTEP_{k,p} + \sum_q FEC_{k,q}\, CTEC_{k,q} = TC_k \quad \forall k \qquad \text{Equation 12}$$

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 13. This ensures that a decision taken to CCS retrofit power plant *i* in period *k* would not be reversed in the subsequent periods.

$$(FR_i)_{k+1} \geq (FR_i)_k \qquad k = 1,2,\dots,n-1 \qquad \text{Equation 13}$$

The constraints regarding the total CO_2 emissions and total energy costs in period *k* are presented in Equation 14 and Equation 15 respectively.

$$TE_k = L_k \qquad \forall k \qquad \text{Equation 14}$$

$$TC_k \leq BD_k \qquad \forall k \qquad \text{Equation 15}$$

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 16 or Equation 17, depending on the user preference. For Equation 16, the total energy cost is minimised subject to constraints in Equation 1 till Equation 14. 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 17, the total emission is minimised subject to the constraints in Equation 1 till Equation 13, and Equation 15. 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 '**B27**' in the worksheet tab '**PLANT_DATA**'. There are two optimisation choices provided i.e., '**min_budget**' (Equation 16) and '**min_emission**' (Equation 17).

$$min\, TC_k \qquad \forall k \qquad \text{Equation 16}$$

$$min\, TE_k \qquad \forall k \qquad \text{Equation 17}$$

The resulting mathematical formulation is a mixed-integer linear programming (MILP) model. The model is implemented in Pyomo, with an easy-to-use input spreadsheet to formulate one's problems. 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.

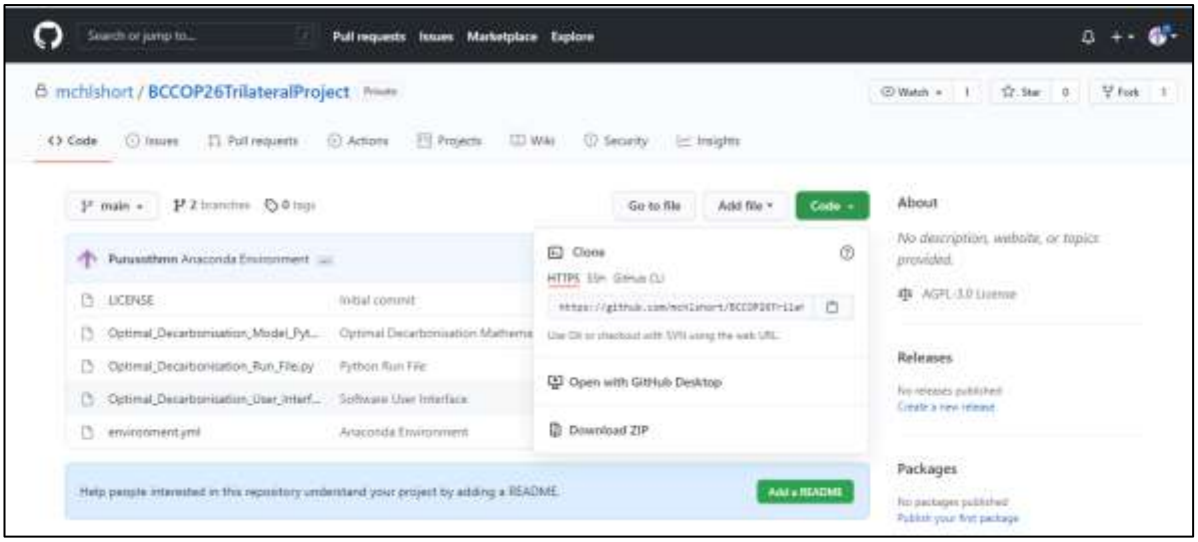


Figure 1: GitHub Landing Page

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 MILP model is solved using **Pyomo, a Python-based open-source optimisation modelling language**. To perform the optimisation task in Pyomo, there are two parts of installation necessary beforehand i.e., **Anaconda and Octeract Engine Solver**. The following section describes the installation procedure of Anaconda on a user's operating machine. Note that some users might experience issues during the installation of both programmes. If any issue occurs, a user should choose to 'run as admin' before installation.

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:

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

ii. During installation, a user should uncheck the 'Add Anaconda3 to my PATH environment variable' box and check the 'Register Anaconda3 as my default Python 3.8' box, as shown in Figure 2.

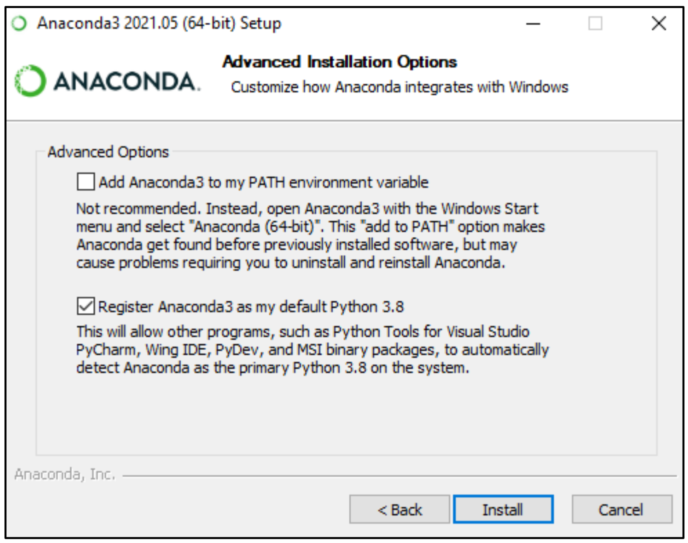


Figure 2: Anaconda Installation Options

iii. Once download, a user should launch the Anaconda application and view the home page as shown in Figure 3.

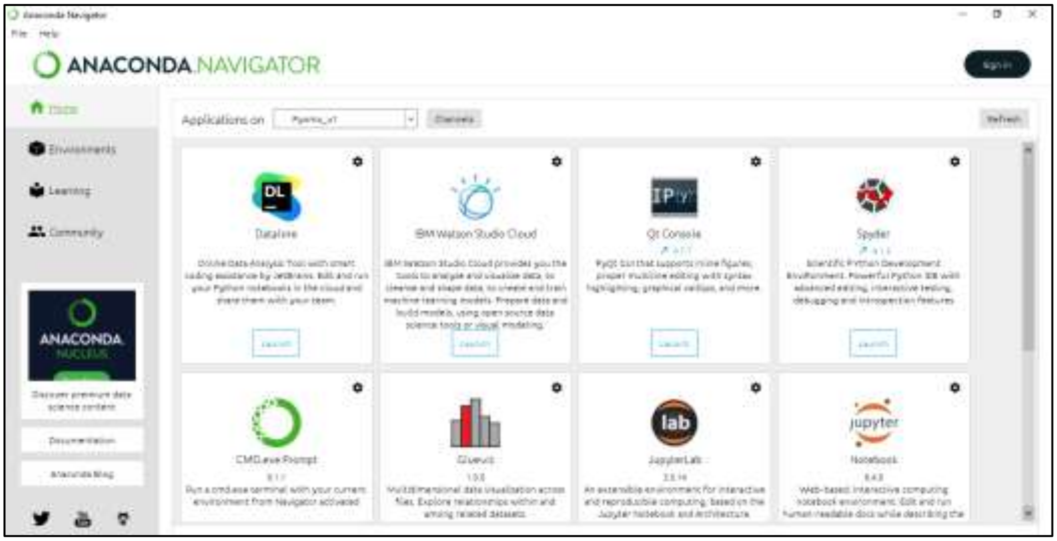


Figure 3: Anaconda Home Page

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

v. At the bottom of the '**Environment**' tab, a user would see four options provided to create a new environment in Anaconda. (see Figure 4)

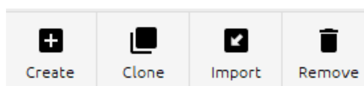


Figure 4: Options for new Anaconda Environment

vi. A user should click on '**Import**' and arrive at the '**Import new environment**' pop-up window visualised in Figure 5.

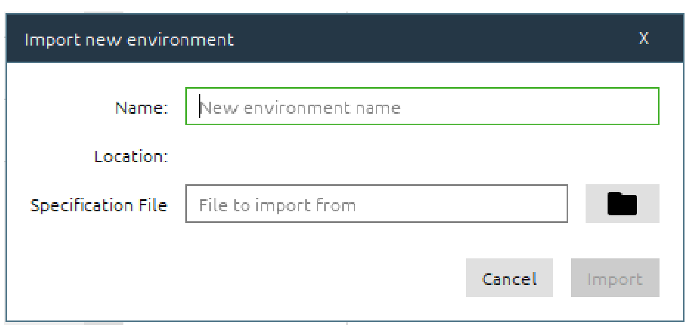


Figure 5: Import New Environment

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

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

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

ix. 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.

x. 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.

xi. Next, a user should left click on the arrow and select '**Open Terminal**', as shown in Figure 6.

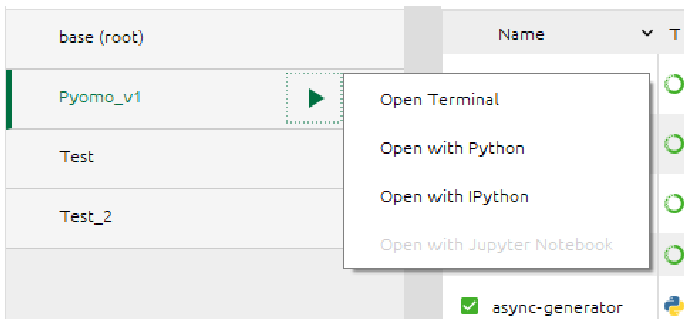


Figure 6: Accessing Environment Terminal

xii. A pop-up window would appear as shown in Figure 7. 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 7).

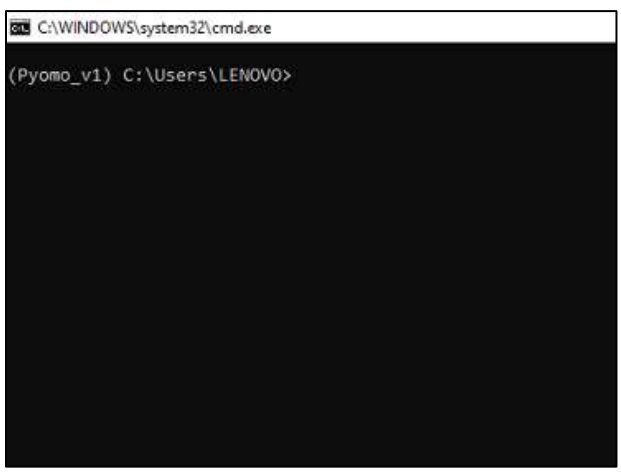


Figure 7: Environment Terminal Command Prompt

xiii. Under the terminal, a user should type '**Spyder**' and click enter, as shown in Figure 8.

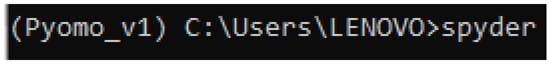


Figure 8: Launching Spyder from Environment Terminal

xiv. The Spyder application would be launched, and a user would be able to view the Spyder console, as shown in Figure 9.

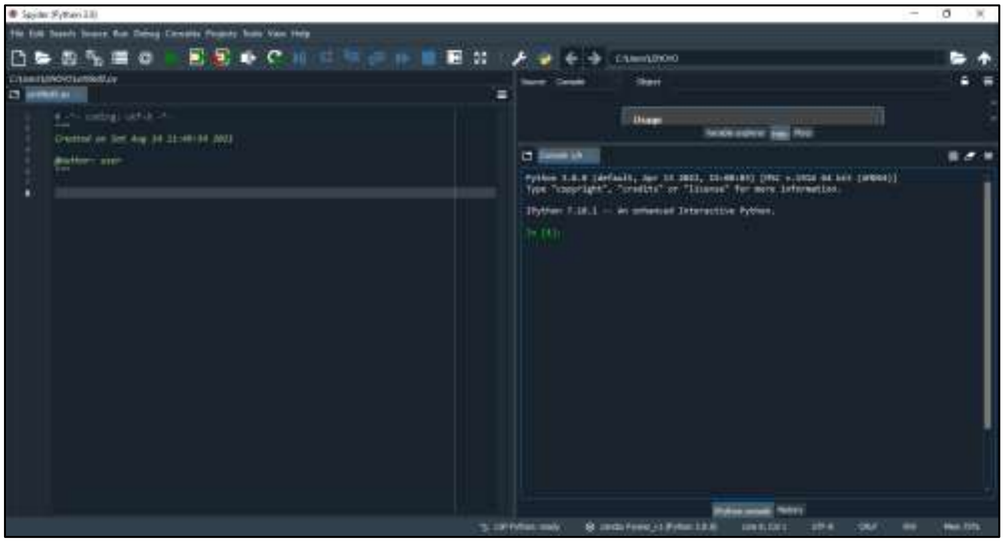


Figure 9: Spyder Console

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 a MILP 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 MILP 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:

i. A user should install the Octeract Engine solver from https://octeract.com/#download
ii. 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.
iii. During installation, the user would be prompted to choose an authentication token, as shown in Figure 10.
iv. The user should choose the second option '**Use an existing authentication token**' (see Figure 10).
v. The user would then be prompted to select a file containing the authentication token downloaded from the website upon completion of registration.

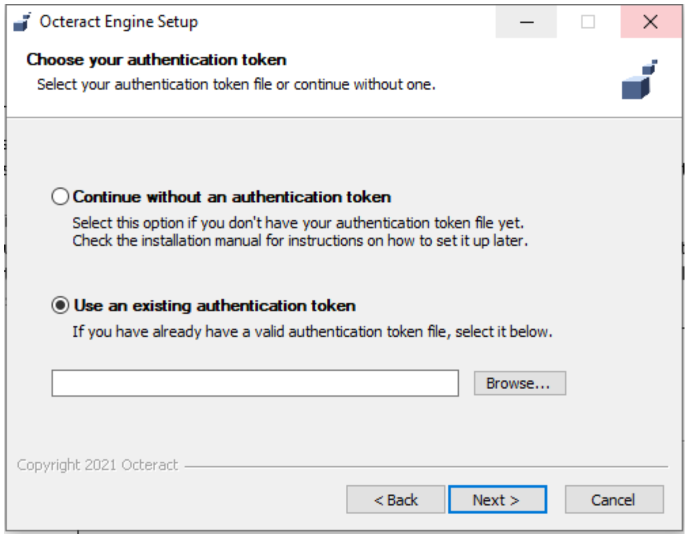


Figure 10: Octeract Engine Authentication Token

vi. Note, the Octeract Engine solver should be installed in the same directory (folder) that was used to save the optimisation files downloaded from GitHub (see Figure 11).

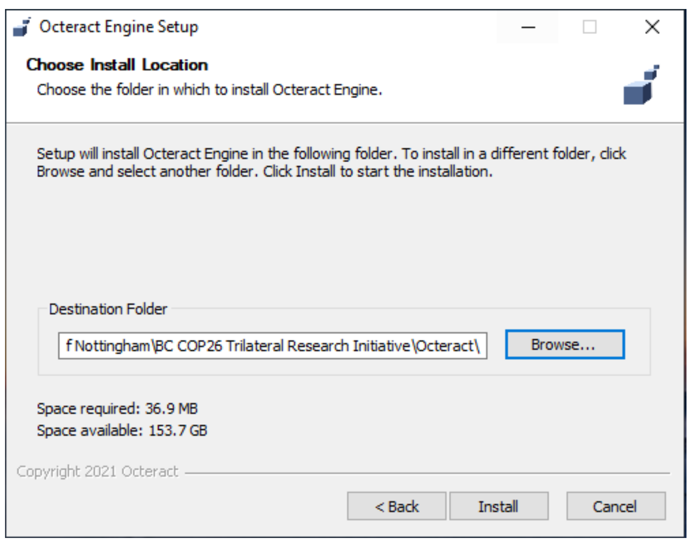
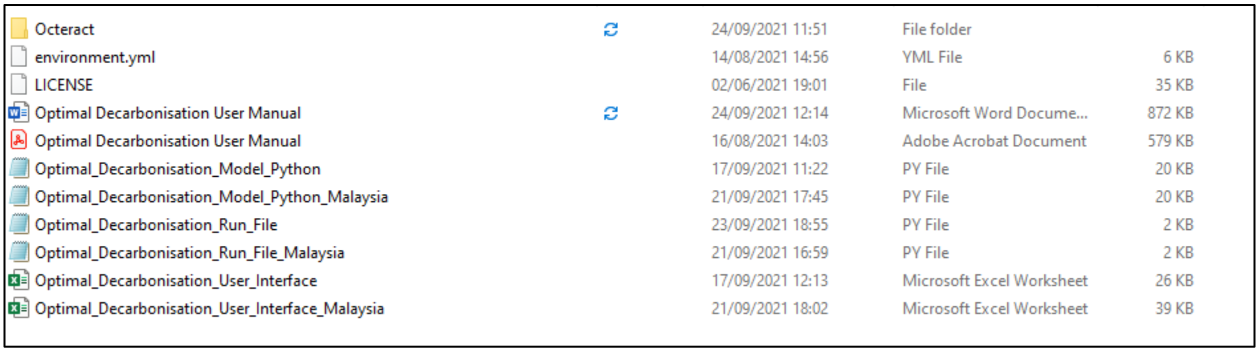


Figure 11: Octeract Engine Solver Directory Change

vii. Upon installation, a user should re-check that the Octeract Engine solver is installed within the same directory as the optimisation files (see Figure 12).



Octeract	24/09/2021 11:51	File folder	
environment.yml	14/08/2021 14:56	YML File	6 KB
LICENSE	02/06/2021 19:01	File	35 KB
Optimal Decarbonisation User Manual	24/09/2021 12:14	Microsoft Word Docume...	872 KB
Optimal Decarbonisation User Manual	16/08/2021 14:03	Adobe Acrobat Document	579 KB
Optimal_Decarbonisation_Model_Python	17/09/2021 11:22	PY File	20 KB
Optimal_Decarbonisation_Model_Python_Malaysia	21/09/2021 17:45	PY File	20 KB
Optimal_Decarbonisation_Run_File	23/09/2021 18:55	PY File	2 KB
Optimal_Decarbonisation_Run_File_Malaysia	21/09/2021 16:59	PY File	2 KB
Optimal_Decarbonisation_User_Interface	17/09/2021 12:13	Microsoft Excel Worksheet	26 KB
Optimal_Decarbonisation_User_Interface_Malaysia	21/09/2021 18:02	Microsoft Excel Worksheet	39 KB

Figure 12: Octeract Engine Solver Directory

viii. For a smoother user experience, a user should set the path directory of the environment variables. A user should search for ‘**environment variables**’ (see Figure 13) and arrive at the page shown in Figure 14.

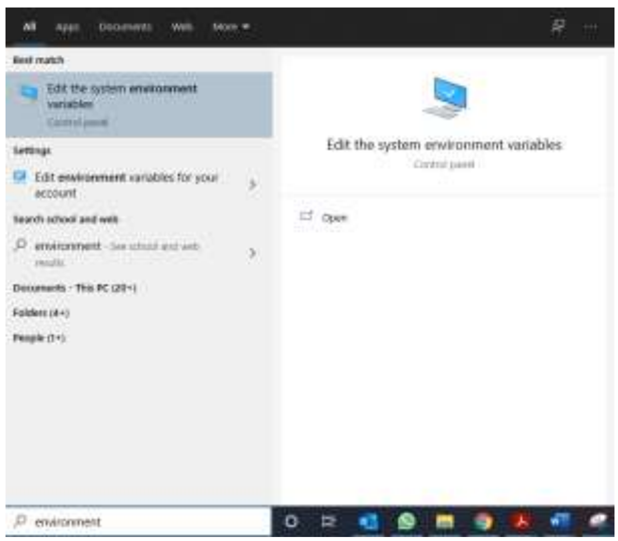


Figure 13: Environment Variable Search from Start-up Menu

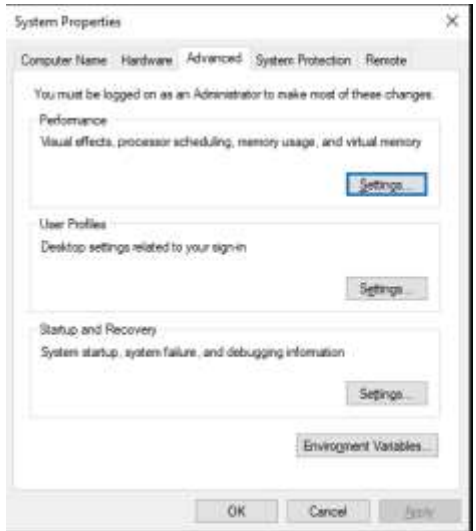


Figure 14: System Properties

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

x. Under the ‘user variables for user’ box, a user should select ‘**Path**’ and click ‘**Edit**’, as shown in Figure 15.

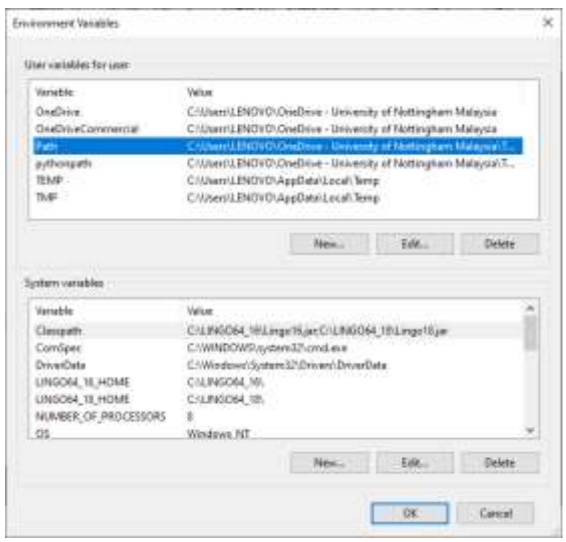


Figure 15: Environment Variables

xi. Then, a user should click ‘**New**’ to add the directory that the necessary files for the optimal decarbonisation software framework (see Figure 16). The directory should be similar to the pathname for the optimisation files (see Figure 17). Note that a user only needs to add one directory.

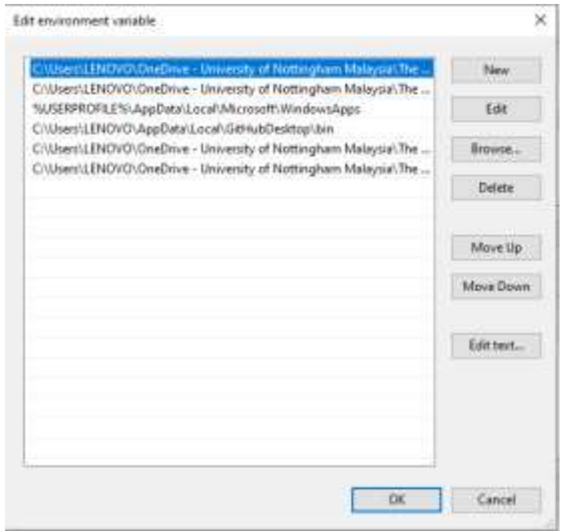


Figure 16: Edit Environment Variables

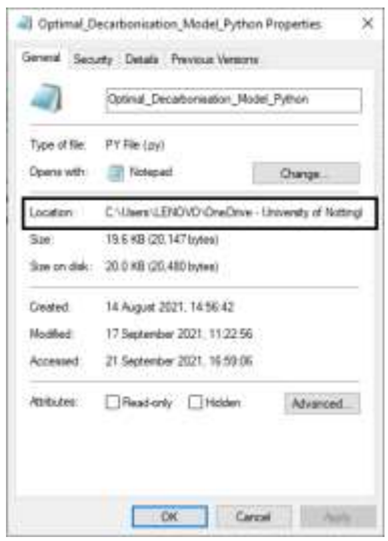


Figure 17: Pathname of Optimisation Files

xii. For more information regarding environment variables and Octeract Engine installation, kindly refer to the explanatory video and installation guide respectively.

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, the pathname in line 14 must be changed.
iii. To obtain the pathname, a user would right-click the folder that contains the optimisation files and click '**Properties**'. Under the '**General**' tab, the pathname could be found at the '**Location**' (see Figure 18).
iv. Once the pathname had been copied, a user should paste and replace the text *"COPY_PATHNAME_ AND_REPLACE_THIS_TEXT"* in line 14 of the '**Optimal_Decarbonisation_Run_File**' file. Note that the pathname must be placed within **single quotation marks**.

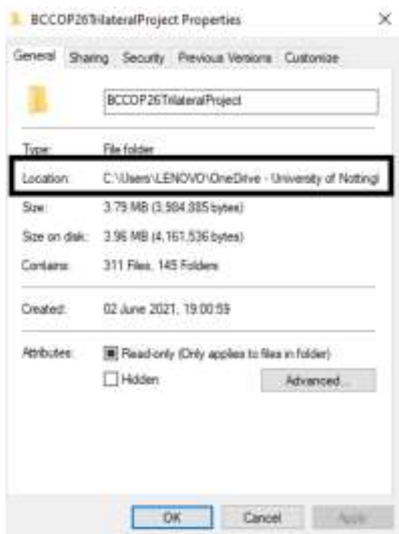


Figure 18: File Directory

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 save and 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 19.

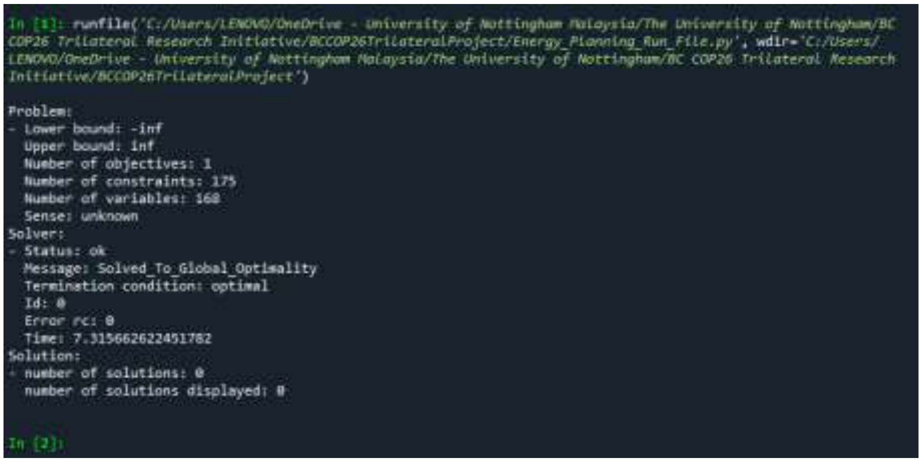


Figure 19: Solver Status

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 '**COST_NET_DATA**' tab consisting of the optimisation results for each period.
x. Note that a user should consider deleting the results sheet or moving the results to a separate Microsoft Excel file before optimising a different case study. Else, the results would be duplicated, thus messy.

The next section presents the base case study to demonstrate the optimisation of the decarbonisation framework via Python.

3. Base Case Study

The application of the optimal decarbonisation software framework is demonstrated with a hypothetical case study. In the base 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 technologies differ in terms of their removal ratios, parasitic power losses and costs. 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 MILP 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 the base case study are presented in Table 4.

Table 4: Power Plant Data for Base Case Study

	Plant 1	Plant 2	Plant 3	Plant 4	Plant 5	Plant 6	Plant 7	Plant 8
Fuel type	REN	Natural Gas	Natural Gas	Natural Gas	Oil	Coal	Coal	Coal
$F_{i,LB}$ / TWh y^{-1}	10	0	0	0	0	0	0	0
$F_{i,UB}$ / TWh y^{-1}	40	40	40	40	10	20	20	20
CS_i / Mt TWh^{-1}	0	0.5	0.5	0.5	0.8	1.0	1.0	1.0

Based on Table 4, 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. A snapshot of the plant data in the Microsoft Excel file titled '**Optimal_Decarbonisation_User_Interface**' is shown in Figure 20. 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

Figure 20: Plant Data in the User Interface

3.2. Energy Planning Data

The energy planning data for the base case study are presented in Table 5.

Table 5: Energy Planning Data for Base Case Study

Period	D_k / TWh y^{-1}	L_k/ Mt y^{-1}	BD_k /mil USD y^{-1}	$CIC_{k,1}$ / Mt TWh^{-1}	$CTC_{k,1}$ /mil RM y^{-1}	$CIC_{k,2}$ / Mt TWh^{-1}	$CTC_{k,2}$ /mil RM y^{-1}
1	60	15	1,500	0	40	0.15	20
2	75	8	2,000	0	40	0.15	20
3	90	3	2,500	0	40	0.15	20

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, there are two types of compensatory energy available for us. The first type has a zero-carbon intensity with a cost of USD 38 mil y^{-1}. The second type is cheaper at USD 20 mil y^{-1}, at the expense of a positive carbon intensity i.e., 0.15 Mt TWh^{-1}. Note that the energy planning data should be inputted in the worksheet tab titled '**EP_DATA**'.

3.3. Energy Data

The energy data for the base case study are presented in Table 6.

Table 6: Energy Data for Base Case Study

Period	Renewable / mil USD TWh^{-1}	Natural Gas / mil USD TWh^{-1}	Oil / mil USD TWh^{-1}	Coal / mil USD TWh^{-1}
1	38	25	49	12
2	38	25	49	12
3	38	25	49	12

Based on Table 6, coal is the cheapest energy source i.e., USD 12 mil TWh^{-1}, while oil is the most expensive i.e., USD 49 mil TWh^{-1}. Note that the energy data should be inputted in the worksheet tab titled '**ENERGY_DATA**'.

3.4. Alternative Fuel Data

The data for the alternative solid and gas fuels are presented in Table 7 and Table 8 respectively.

Table 7: Alternative Solid Fuel Data

Period	$CIAS_{i,k,1}$	$CTAS_{i,k,1}$	$CIAS_{i,k,2}$	$CTAS_{i,k,2}$
1	0.2	20	0.4	15
2	0.2	20	0.4	15
3	0.2	20	0.4	15

Table 8: Alternative Gas Fuel Data

Period	$CIAG_{i,k,1}$	$CTAS_{i,k,1}$	$CIAS_{i,k,2}$	$CTAS_{i,k,2}$
1	0.15	35	0.25	30
2	0.15	35	0.25	30
3	0.15	35	0.25	30

Based on Table 7 and Table 8, there are two alternatives for solid and gas fuels, which could potentially substitute the use of fossil-based fuels i.e., coal and natural gas. Each alternative fuel has its respective carbon intensity and cost. Note that the data associated with the alternative solid and gas fuel should be inputted in the worksheet tab titled '**ALT_SOLID**' and '**ALT_GAS**' respectively.

3.5. CCS Data

The CCS data for the base case study are presented in Table 9.

Table 9: CCS Data for Base Case Study

Period	$RR_{1,k}$	$X_{1,k}$	$CTR_{i,k,1}$ / mil USD y^{-1}	$CFX_{i,k,1}$ / mil USD y^{-1}	$RR_{2,k}$	$X_{2,k}$	$CTR_{i,k,2}$ / mil USD y^{-1}	$CFX_{i,k,2}$ / mil USD y^{-1}
1	0.85	0.15	34	200	0.65	0.25	29	150
2	0.88	0.13	32	180	0.68	0.23	27	130
3	0.90	0.10	30	160	0.70	0.20	25	110

Based on Table 9, there are two CCS technologies available for deployment in the base case study. 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. Across the periods, the parasitic power loss and costs for both CCS technologies decrease while the removal ratios increase. In other words, both CCS technologies are more effective for CO_2 load removal across the periods. Note that the CCS data should be inputted in the worksheet tab titled '**CCS_DATA**'.

3.6. NETs Data

The NETs data for the base case study are presented in Table 10 and Table 11.

Table 10: NETs Carbon Intensity Data for Base Case Study

Period	$CIEP_{k,1}$ / Mt TWh^{-1}	$CIEP_{k,2}$ / Mt TWh^{-1}	$CIEP_{k,3}$ / Mt TWh^{-1}	$CIEC_{k,2}$ / Mt TWh^{-1}	$CIEC_{k,2}$ / Mt TWh^{-1}	$CIEC_{k,2}$ / Mt TWh^{-1}
1	-0.8	-0.6	-0.4	-0.6	-0.4	-0.2
2	-0.8	-0.6	-0.4	-0.6	-0.4	-0.2
3	-0.8	-0.6	-0.4	-0.6	-0.4	-0.2

Table 11: NETs Cost Data for Base Case Study

Period	$CTEP_{k,1}$ / mil USD y^{-1}	$CTEP_{k,2}$ / mil USD y^{-1}	$CTEP_{k,3}$ / mil USD y^{-1}	$CTEC_{k,1}$ / mil USD y^{-1}	$CTEC_{k,2}$ / mil USD y^{-1}	$CTEC_{k,3}$ / mil USD y^{-1}
1	43	40	37	49	37	24
2	43	40	37	49	37	24
3	43	40	37	49	37	24

Based on Table 10 and Table 11, note that there are three technology choices of EP-NETs and EC-NETs available for deployment. Each NETs vary in terms of carbon intensity and cost. Note that NETs with the highest carbon intensity is the most expensive. Note that the carbon intensity and cost data for NETs should be inputted in the worksheet tab titled '**CI_NET_DATA**' and '**COST_NET_DATA**' respectively.

3.7. Optimisation Results

The optimisation of the base case study 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 1255, 1988 and 2584 mil USD y^{-1} respectively**. For period 1, only CCS technology 2 is deployed for natural gas-based power plant 2.

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 8.2, 7.8 and 4.7 Mt y^{-1} respectively**. In other words, the total CO_2 emissions for all periods are higher than the CO_2 emission limit. Figure 21 presents a sample snapshot for the minimum budget optimisation results for period 1.

	Fuel	Energy	CI	CS_1 Selectio	CS_2 Selectio	CCS_1 Ret	CCS_2 Ret	SOLID_1	SOLID_2	GAS_1	GAS_2	Net Energy	Carbon Load	Cost
Plant_1	REN	10	0	0	0	0	0	0	0	0	0	10	0	
Plant_2	NG	0	0.5	0	0	0	0	0	0	0	0	0	0	
Plant_3	NG	0	0.5	0	0	0	0	0	0	0	0	0	0	
Plant_4	NG	0	0.5	0	0	0	0	0	0	0	0	0	0	
Plant_5	OIL	0	0.8	0	0	0	0	0	0	0	0	0	0	
Plant_6	COAL	10	1	0	0	0	0	5	5	0	0	0	0	
Plant_7	COAL	20	1	0	0	0	0	0	20	0	0	0	0	
Plant_8	COAL	20	1	0	0	0	0	20	0	0	0	0	0	
EP_NET_1	EP_NET_1		-0.8									0	0	
EP_NET_2	EP_NET_2		-0.6									0	0	
EP_NET_3	EP_NET_3		-0.4									0	0	
EC_NET_1	EC_NET_1		-0.6									0	0	
EC_NET_2	EC_NET_2		-0.4									0	0	
EC_NET_3	EC_NET_3		-0.2									0	0	
COMP_1	COMP_1		0									0	0	
COMP_2	COMP_2		0.15									0	0	
TOTAL													15	1,255.00

Figure 21: Minimum Budget Optimisation Results for Period 1

Based on Figure 21, 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. Note that a user may observe a different set of solutions (degenerate solutions). In this case, a user should check if the objective functions stay the same.

4. Potential Bugs and Solutions

The optimisation of the decarbonisation framework would take place in Pyomo while utilising the Octeract Engine as the choice of the solver. During the optimisation procedure, a user may encounter several potential bugs. Therefore, this section aims to provide solutions for some of the potential bugs that may be encountered by a user.

4.1. Excel Permission Denied

For this error, a user might obtain an error message as shown in Figure 22.

```
PermissionError: [Errno 13] Permission denied:
'Optimal_Decarbonisation_User_Interface.xlsx'
```

Figure 22: Excel Permission Denied Error

To avoid this error, a user should save and close the '**Optimal_Decarbonisation_User_Interface' Excel** file before running the file in Python/Sypder. Upon running the Python file, a user may re-open the Excel file to view the results.

4.2. Absent Pyomo Module

For this error, a user might obtain an error message as shown in.

```
ModuleNotFoundError: No module named 'pyomo'
```

Figure 23: Absent Module Error

To overcome this error, a user should ensure that the Pyomo environment is properly installed within the selected environment in Anaconda. Note that the ‘**environment.yml**’ file contains the Pyomo environment. Therefore, importing the ‘**environment.yml**’ file (see Section 2.6) during Anaconda installation would overcome this issue.