Storage Optimisation in Automated Fulfilment Centres

Technical and User Guides

Team MCS15

Cheryl Frances Lee
Project Manager
clee0070@student.monash.edu

Jonathan How Yu Chern
Technical Lead
yhow0004@student.monash.edu

Anson Sameer Lee
Quality Assurance
alee0061@student.monash.edu

1. Technical guide

This section contains instructions on how to set up the requisite environment to run the program on your device.

1.1 Software requirements

Operating system	Windows 10, Windows 11
IDE	Visual Studio Code
Python version	Python 3.9.0 minimum
Other	Git installed on device, GitHub account required

Compatibility with alternate software and operating systems has not been verified. As such, it is recommended that the program is run only within the environment specified above.

1.2 Installing additional libraries

The program relies on the NumPy and MatPlotLib libraries to produce three-dimensional renders of its solutions. These libraries must be installed prior to running the program.

Assuming Python has been installed, the pip package manager can be installed via the command line as follows (if it is not already installed):

```
python -m pip install -U pip
```

The requisite libraries can be installed with the following commands:

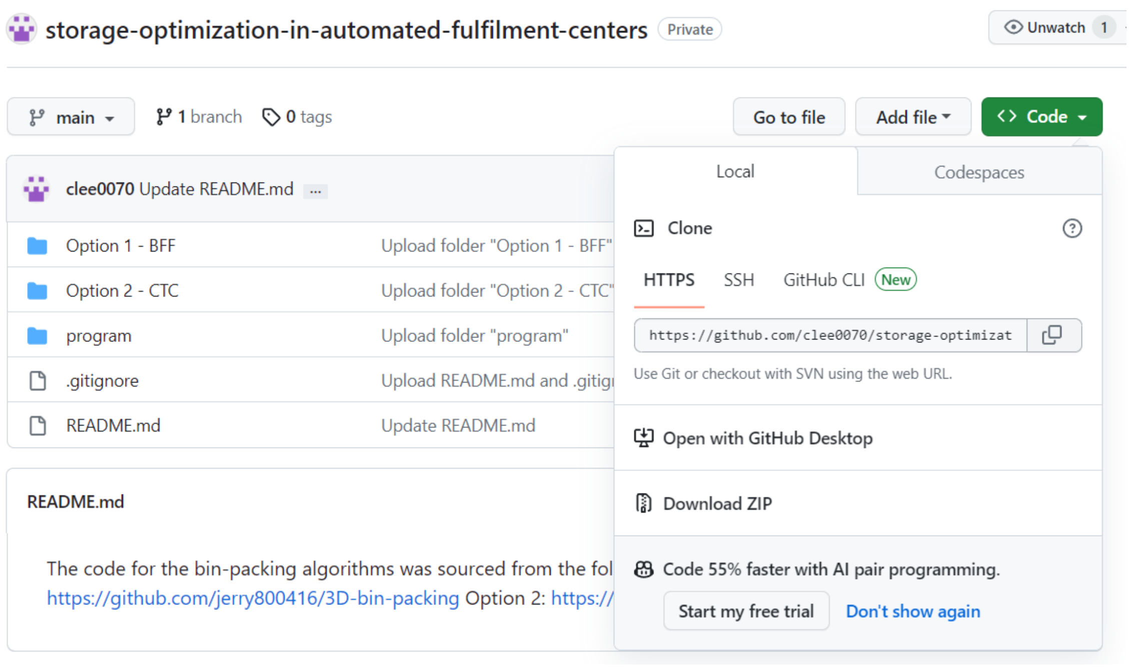
```
python -m pip install -U numpy

python -m pip install -U matplotlib
```

1.3 Cloning the repository

Ensure the requisite software has been installed (see **Software requirements**) and you are connected to the Internet prior to cloning the repository.

1. Navigate to the repository on GitHub: clee0070/storage-optimization-in-automated-fulfilment-centers (github.com)

2. Clicking on the *"<> Code"* button reveals a menu with several options for cloning the repository. Click on the button next to the URL to copy it to the clipboard. Cloning via HTTPS is recommended as issues were reported when attempting to clone via SSH.



clee0070 Update README.md ...	
Option 1 - BFF	Upload folder "Option 1 - BFF"
Option 2 - CTC	Upload folder "Option 2 - CTC"
program	Upload folder "program"
.gitignore	Upload README.md and .gitig
README.md	Update README.md

README.md

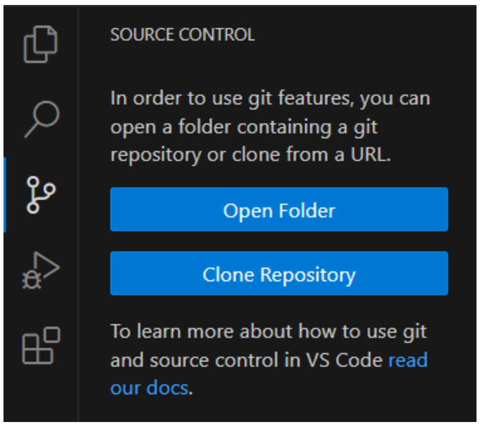
The code for the bin-packing algorithms was sourced from the fol
https://github.com/jerry800416/3D-bin-packing Option 2: https://

```
https://github.com/clee0070/storage-optimizat
```

Use Git or checkout with SVN using the web URL.

Code 55% faster with AI pair programming.

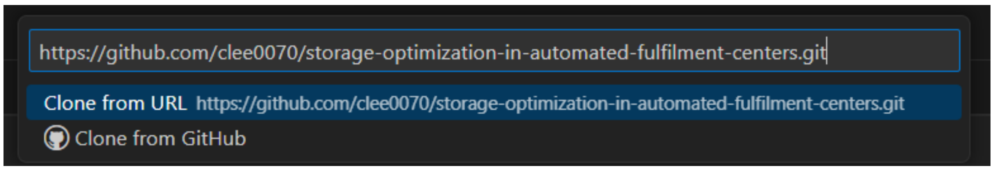
3. Open a new window in VS code and navigate to the source control menu. Click on "*Clone Repository*".



In order to use git features, you can open a folder containing a git repository or clone from a URL.

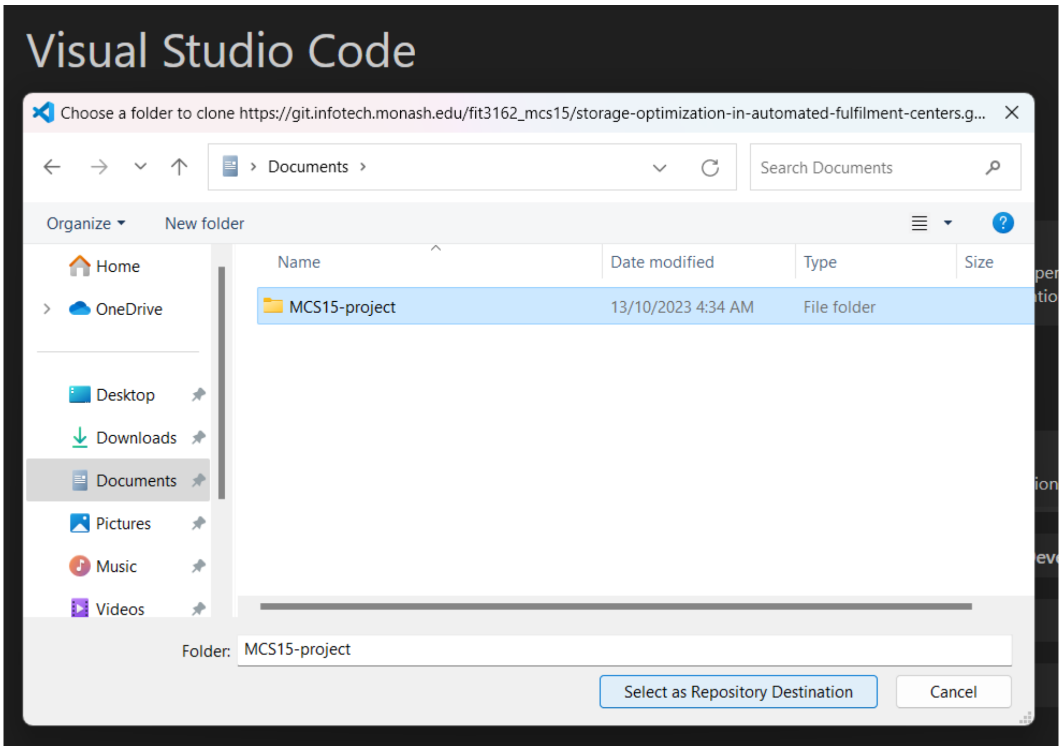
To learn more about how to use git and source control in VS Code read our docs.

Paste the URL in the text box at the top of the screen and press *Enter*. Ensure *"Clone from URL"* is highlighted.



```
https://github.com/clee0070/storage-optimization-in-automated-fulfilment-centers.git
Clone from URL  https://github.com/clee0070/storage-optimization-in-automated-fulfilment-centers.git
Clone from GitHub
```

4. When the file explorer appears, select the directory you wish to clone the repository to. It is recommended to use a new empty directory for this purpose.



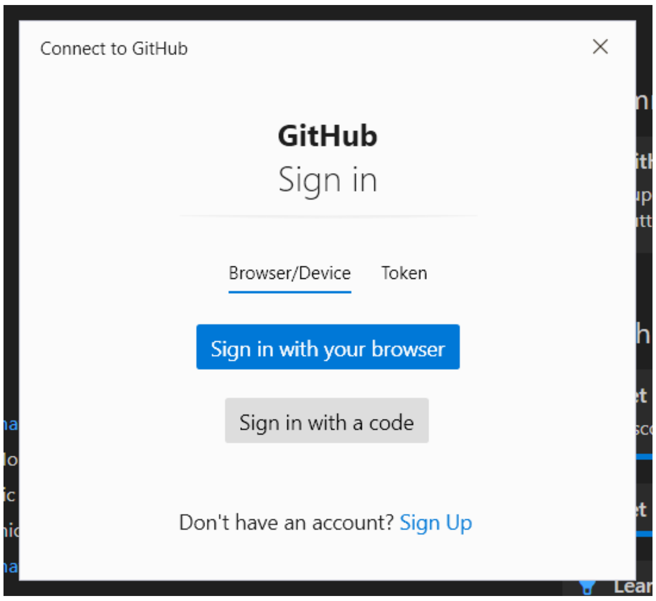
Visual Studio Code

Choose a folder to clone https://git.infotech.monash.edu/fit3162_mcs15/storage-optimization-in-automated-fulfilment-centers.g...

Name	Date modified	Type	Size
MCS15-project	13/10/2023 4:34 AM	File folder	

Folder: MCS15-project

5. You will be prompted to sign in to your GitHub account. Click on *"Sign in with your browser"*.



Connect to GitHub

GitHub

Sign in

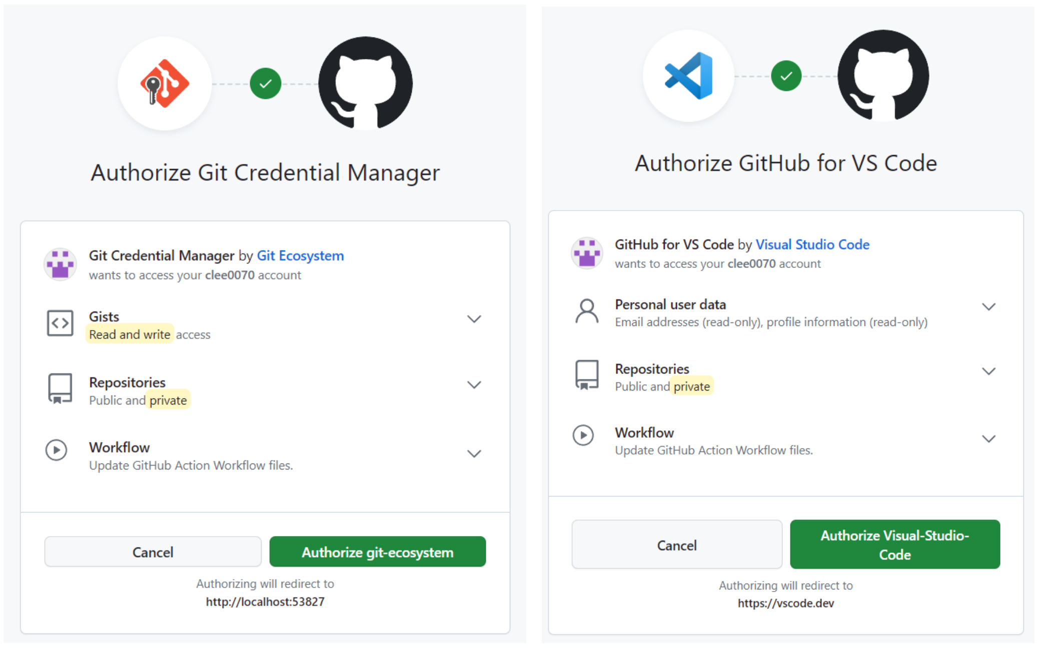
Browser/Device Token

Sign in with your browser

Sign in with a code

Don't have an account? Sign Up

You will be redirected to your browser and asked to authorise the git credential manager. Click on the green button to proceed. If you are also asked to authorise GitHub for VS Code, do the same.



Authorize Git Credential Manager

Git Credential Manager by Git Ecosystem
wants to access your clee0070 account

Gists
Read and write access

Repositories
Public and private

Workflow
Update GitHub Action Workflow files.

Cancel | Authorize git-ecosystem

Authorizing will redirect to
http://localhost:53827

Authorize GitHub for VS Code

GitHub for VS Code by Visual Studio Code
wants to access your clee0070 account

Personal user data
Email addresses (read-only), profile information (read-only)

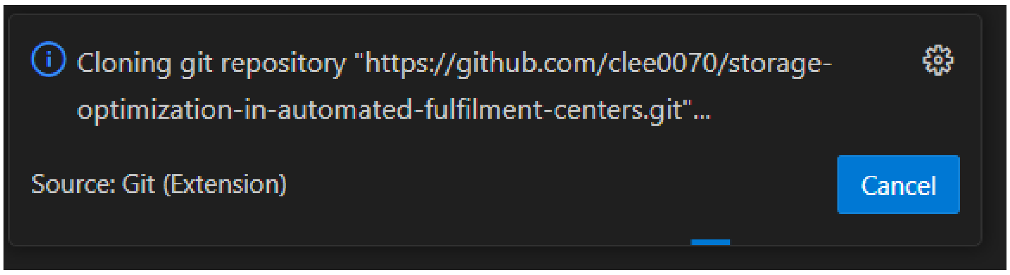
Repositories
Public and private

Workflow
Update GitHub Action Workflow files.

Cancel | Authorize Visual-Studio-Code

Authorizing will redirect to
https://vscode.dev

6. Wait until cloning is finished. The duration may vary depending on your connection speed.

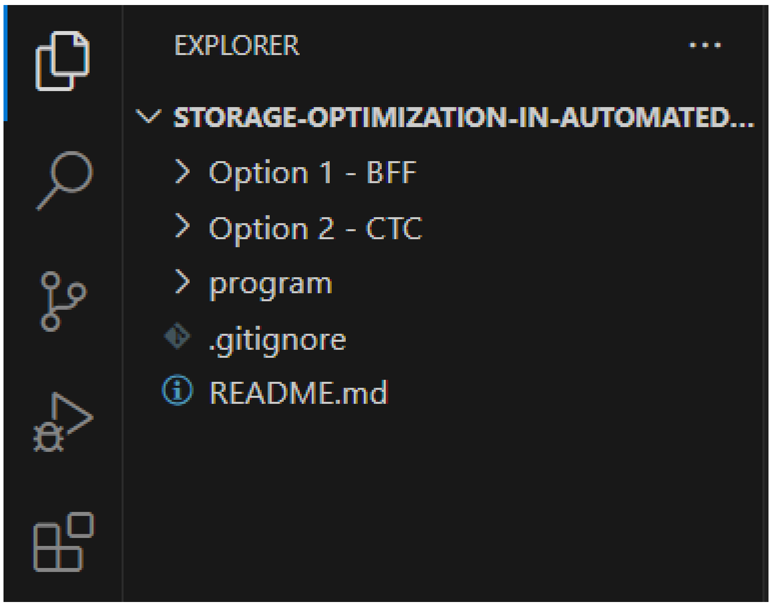


Cloning git repository "https://github.com/clee0070/storage-optimization-in-automated-fulfilment-centers.git"...

Source: Git (Extension)

Cancel

Once cloning is finished, navigate to the directory the repository was cloned to or open it in VS Code's file explorer. It should contain the following files and folders:


EXPLORER

STORAGE-OPTIMIZATION-IN-AUTOMATED...
- Option 1 - BFF
- Option 2 - CTC
- program
- .gitignore
- README.md

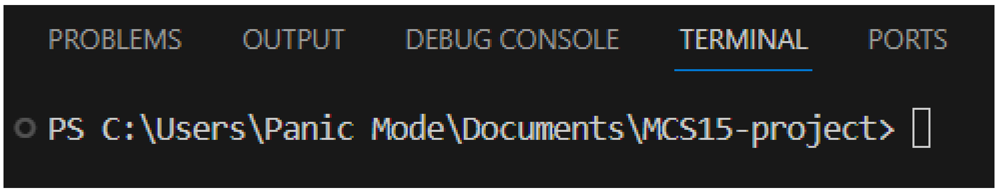
2. User guide

2.1 Managing input and output files

Additional files and directories will be generated over the course of the program's execution. This section details the purpose of these files and contains instructions on how to manage them.

2.1.1 Where are generated files located?

When you choose to generate your first input file, the directories containing it and all future additional files will be created within the current directory. The current directory is indicated via the terminal in VS code:



```
PS C:\Users\Panic Mode\Documents\MCS15-project>
```

If you wish for these directories to be created elsewhere, you can change the current directory with the following command:

```
cd C:\Users\username\Documents
```

2.1.2 Sample file directory

This diagram illustrates an example directory structure after some input and output files have been generated for both options. The purpose of each directory and type of file will be explained in the following section.

```
CURRENT DIRECTORY

    -- files_Option1                       -- files_Option2

        -- csv_inputs                          -- csv_inputs

            -- fileBinCount.txt                    -- fileBinCount.txt
            -- fileBoxCount.txt                    -- fileBoxCount.txt

            -- csv_bins                            -- csv_bins
                -- inputBins1.csv                      -- inputBins1.csv
                -- inputBins2.csv                      -- inputBins2.csv
                ...                                    ...

            -- csv_boxes                           -- csv_boxes
                -- inputBoxes1.csv                     -- inputBoxes1.csv
                -- inputBoxes2.csv                     -- inputBoxes2.csv
                ...                                    ...

        -- csv_outputs                         -- csv_outputs

            -- fileOutCount.txt                    -- fileOutCount.txt
            -- lastBinFile.txt                     -- lastBinFile.txt
            -- lastBoxFile.txt                     -- lastBoxFile.txt

            -- outputFitted1.csv                   -- outputFitted1.csv
            -- outputFitted2.csv                   -- outputFitted2.csv
            ...                                    ...

            -- outputUnfitted1.csv                 -- outputUnfitted1.csv
            -- outputUnfitted2.csv                 -- outputUnffited2.csv
            ...                                    ...

            -- outputBins1.csv                     -- outputBins1.csv
            -- outputBins2.csv                     -- outputBins2.csv
            ...                                    ...

            -- outputSummary1.csv                  -- outputSummary1.csv
            -- outputSummary2.csv                  -- outputSummary2.csv
            ...
```

2.1.3 Directories

Directory	**Purpose**	**Path**
files_OptionX	“Master” directory containing all the generated input and output files pertaining to a particular option.	cd/files_OptionX
csv_inputs	Contains files and sub-directories used and created by subroutines that generate input files.	cd/files_OptionX/csv_inputs
csv_bins	Contains generated input CSV files where each row contains fields and attributes for a single bin/container.	cd/files_OptionX/csv_inputs/csv_bins
csv_boxes	Contains generated input CSV files where each row contains fields and attributes for a single box/item/SKU.	cd/files_OptionX/csv_inputs/csv_boxes
csv_outputs	Contains files used and created by subroutines that generate output files.	cd/files_OptionX/csv_outputs
KEY cd - current directory X - numeric identifier corresponding to either Option 1 or Option 2		

2.1.4 Input and output files

File	**Purpose**	**Parent directory**
	INPUT FILES	
fileBinCount.txt	Contains a single integer representing the number of input bin CSV files that have been generated that is incremented upon the creation of a new file. The program uses this integer to determine the numeric identifier of the next input bin CSV file to be created, effectively ensuring the creation of a new file does not overwrite an existing one.	csv_inputs
fileBoxCount.txt	Contains a single integer representing the number of input box CSV files that have been generated that is incremented upon the creation of a new file. The program uses this integer to determine the numeric identifier of the next input box CSV file to be created, effectively ensuring the creation of a new file does not overwrite an existing one.	csv_inputs
inputBinsX.csv	Contains rows of data where each row represents a singular bin. Exact contents of fields will vary depending on the implementation of the bin class for a particular algorithm.	csv_bins
inputBoxesX.csv	Contains rows of data where each row represents a singular box. Exact contents of fields will vary depending on the implementation of the box class for a particular algorithm.	csv_boxes
	OUTPUT FILES	
fileOutCount.txt	Contains a single integer representing the number of 'sets' of output CSV files that have been generated that is incremented whenever a packing computation results in additional output files. The program uses this integer to determine the numeric identifier of the next set of output files to be created, effectively ensuring the creation of new files does not overwrite existing ones.	csv_outputs
lastBinFile.txt	Contains the relative path of the last input bin CSV file read from. This data is captured upon first reading the input file and is eventually logged in an output summary file.	csv_outputs
lastBoxFile.txt	Contains the relative path of the last input box CSV file read from. This data is captured upon first reading the input file and is eventually logged in an output summary file.	csv_outputs

outputFittedX.csv	Contains rows of data where each row represents a singular box that was successfully packed in a bin. Listed attributes are identical to the source input box file but the identifier of the host bin has also been included.	csv_outputs
outputUnfittedX.csv	Contains rows of data where each row represents a singular box that was unable to pack into any bin. Listed attributes are identical to the source input box file.	csv_outputs
outputBins.Xcsv	Contains rows of data where each row represents a singular bin. In addition to the attributes listed in the source input bin file, additional data items, such as volume utilisation and other notable metrics have been included.	csv_outputs
outputSummaryX.csv	Contains a single row of data with various metrics relevant to the entire packing computation, such as the name of the source input files, total volume utilised, and volume of unpacked boxes.	csv_outputs
KEY X - numeric identifier of each file		

2.1.5 Regarding numeric identifiers of input and output files

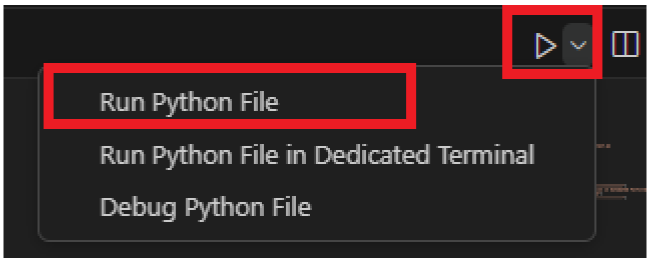
Note that while all output files with the same numeric ID are generated from the same packing computation, they do not necessarily correspond to bin and box input files with the same numeric ID. Furthermore, bin and box input files with identical numeric IDs are not necessarily coupled to each other either. For example *inputBins1.csv* can be used with *inputBoxes2.csv* and the results of the packing may be logged in *outputSummary3.csv*.

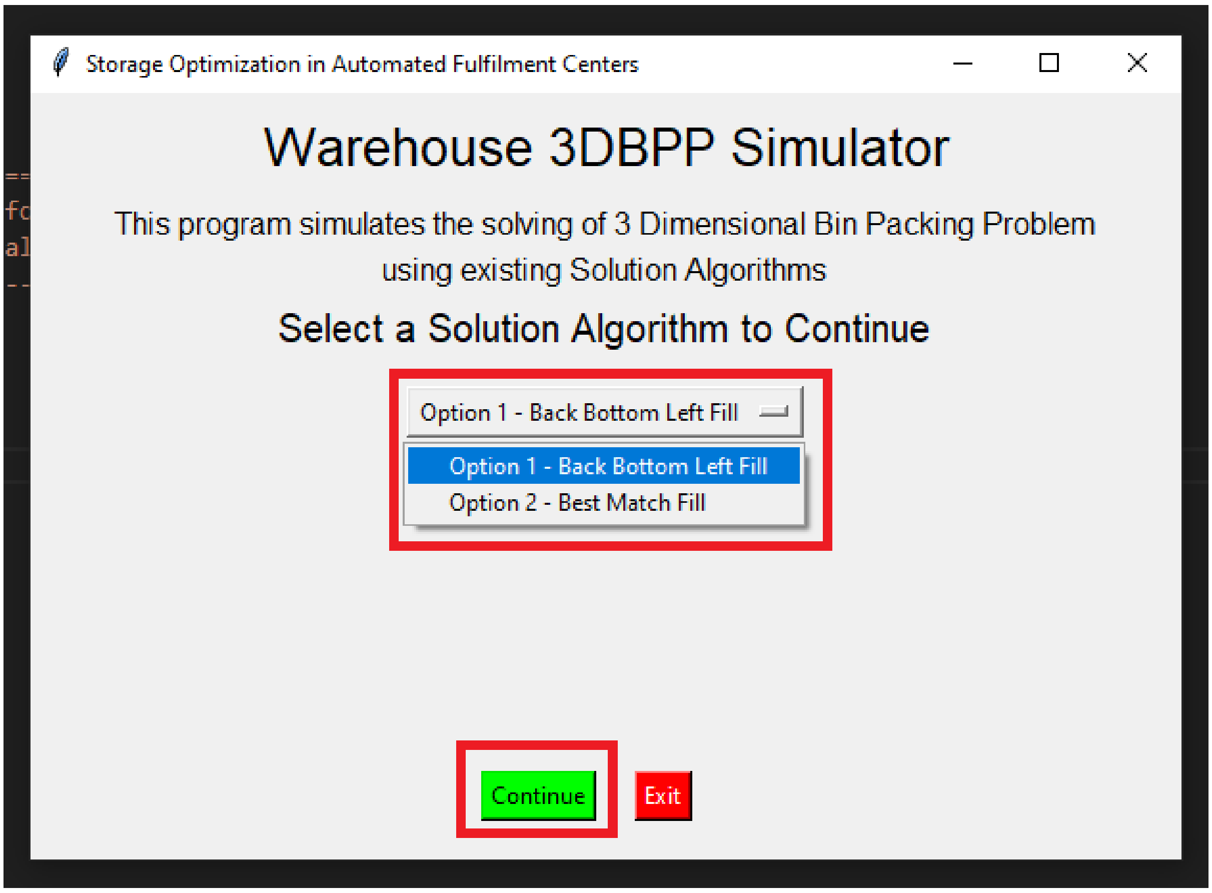
2.1.6 WARNING: Regarding the handling of fileXCount.txt files

As mentioned above, fileBinCount.txt, fileBoxCount.txt, and fileOutCout.txt ensure that each new file generated has a unique name that will not overwrite existing files. For this reason, it is **NOT** recommended to delete or directly modify these files. Doing so may result in the loss of data.

2.2 Running the 3DBPP simulation

To run the simulation, navigate to the main program, **main.py**, and select “Run Python File” on the top right corner of Visual Studio Code.




Storage Optimization in Automated Fulfilment Centers

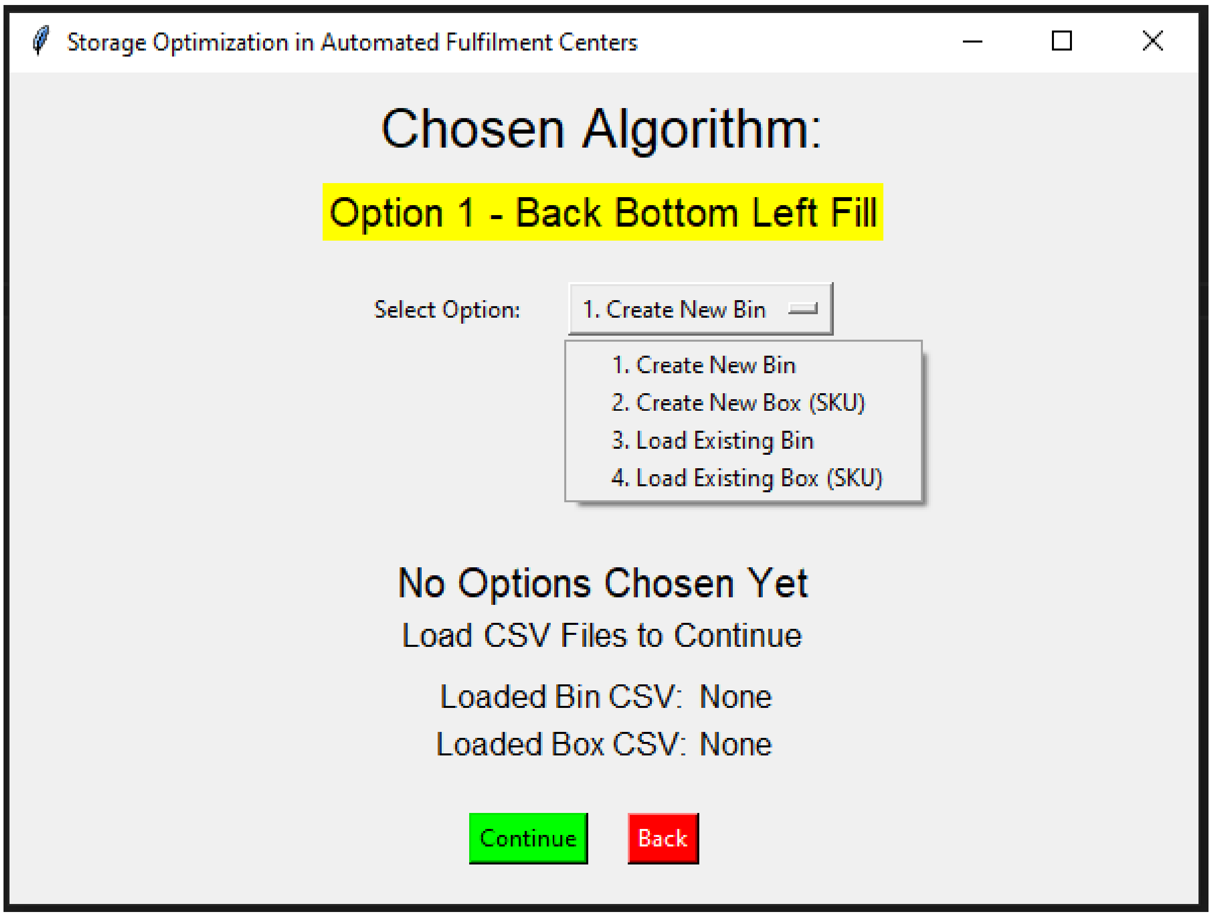
Warehouse 3DBPP Simulator

This program simulates the solving of 3 Dimensional Bin Packing Problem using existing Solution Algorithms

Select a Solution Algorithm to Continue

Option 1 - Back Bottom Left Fill
Option 1 - Back Bottom Left Fill
Option 2 - Best Match Fill
Continue
Exit

On the Main Menu, click on the dropdown menu, and choose the solution algorithm. Then, click on “Continue” to proceed. Otherwise, click on “Exit” to terminate the program.



In the Master Subroutine Window, your current selected solution algorithm is displayed in yellow. You may select one of the 5 options from the dropdown menu:

1. Create New Box (SKU) CSV File
2. Create New Bin CSV File
3. Load Existing Bin CSV File
4. Load Existing Box (SKU) File
5. Compute Algorithm

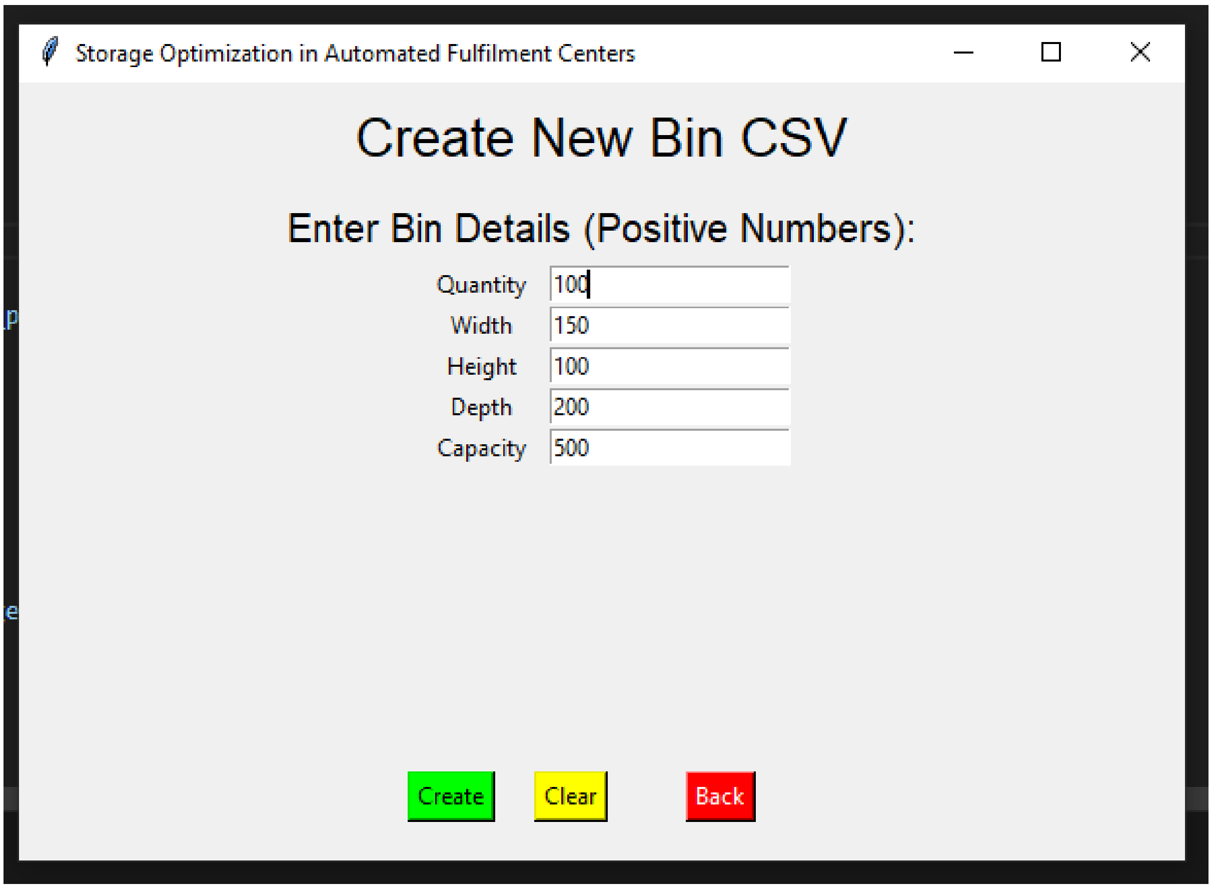
After that, press “Continue” to proceed, or “Back” to return to the main menu if you want to select a different Solution Algorithm.

Note:

- In the beginning, there are no existing Bin or Box CSV files to load. Hence, if there are no CSV files select options 1 and 2 to create them.
- “Compute Algorithm” can only be run if one Bin CSV and one Box CSV have been loaded into the program.

2.2.1 Create new bin CSV file

Upon selecting this option, you will be prompted with the following form. Unlike boxes, the bin entry form is identical for both options.


Storage Optimization in Automated Fulfilment Centers

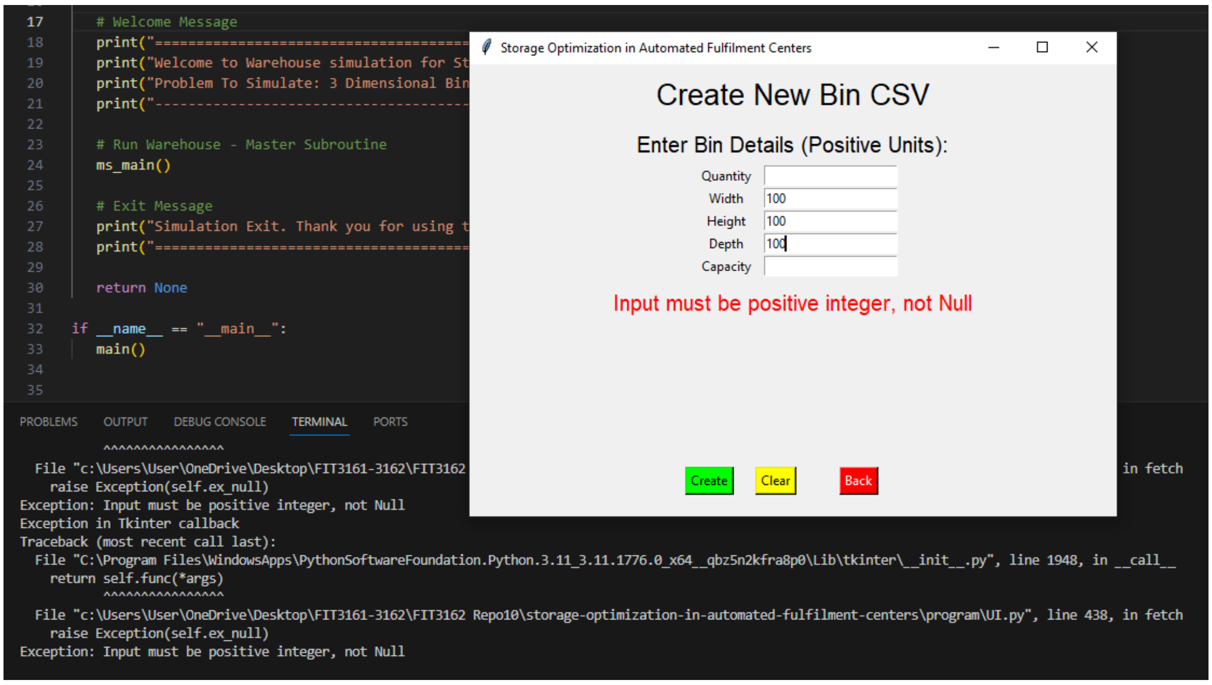
Create New Bin CSV

Enter Bin Details (Positive Numbers):

Quantity	100
Width	150
Height	100
Depth	200
Capacity	500

Attributes for bins are declared explicitly and not randomly generated. All bins will have identical attributes.

- quantity: the number of bins
- width, height, depth: dimensions of the bins
- capacity: maximum weight capacity of a single bin



```
    # Welcome Message
    print("=====================================
    print("Welcome to Warehouse simulation for St
    print("Problem To Simulate: 3 Dimensional Bin
    print("--------------------------------------

    # Run Warehouse - Master Subroutine
    ms_main()

    # Exit Message
    print("Simulation Exit. Thank you for using t
    print("=====================================

    return None

if __name__ == "__main__":
    main()
```

```
    ^^^^^^^^^^^^^^^^
  File "c:\Users\User\OneDrive\Desktop\FIT3161-3162\FIT3162                                  in fetch
    raise Exception(self.ex_null)
Exception: Input must be positive integer, not Null
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.1776.0_x64__qbz5n2kfra8p0\Lib\tkinter\__init__.py", line 1948, in __call__
    return self.func(*args)
           ^^^^^^^^^^^^^^^^
  File "c:\Users\User\OneDrive\Desktop\FIT3161-3162\FIT3162 Repo10\storage-optimization-in-automated-fulfilment-centers\program\UI.py", line 438, in fetch
    raise Exception(self.ex_null)
Exception: Input must be positive integer, not Null
```

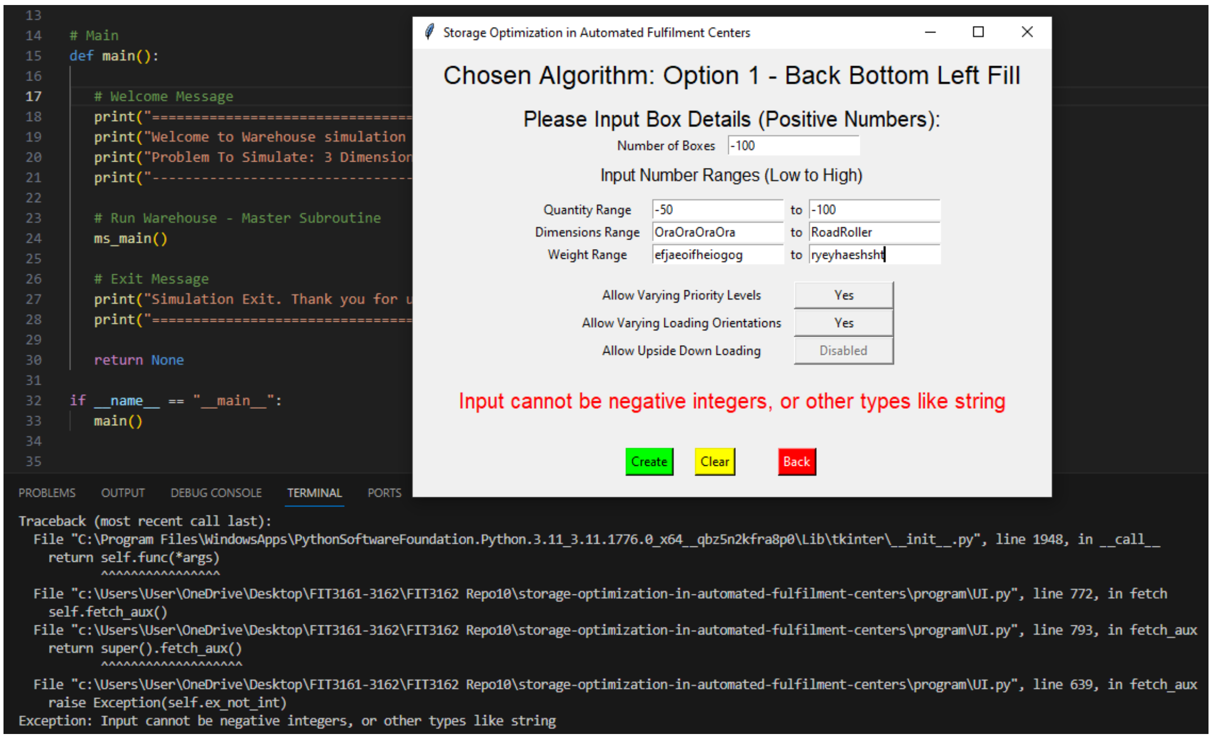
Storage Optimization in Automated Fulfilment Centers

Create New Bin CSV

Enter Bin Details (Positive Units):

Quantity	
Width	100
Height	100
Depth	100
Capacity	

Input must be positive integer, not Null



```
# Main
def main():

    # Welcome Message
    print("=====================================
    print("Welcome to Warehouse simulation
    print("Problem To Simulate: 3 Dimension
    print("--------------------------------------

    # Run Warehouse - Master Subroutine
    ms_main()

    # Exit Message
    print("Simulation Exit. Thank you for u
    print("=====================================

    return None

if __name__ == "__main__":
    main()
```

```
Traceback (most recent call last):
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11_3.11.1776.0_x64__qbz5n2kfra8p0\Lib\tkinter\__init__.py", line 1948, in __call__
    return self.func(*args)
           ^^^^^^^^^^^^^^^^
  File "c:\Users\User\OneDrive\Desktop\FIT3161-3162\FIT3162 Repo10\storage-optimization-in-automated-fulfilment-centers\program\UI.py", line 772, in fetch
    self.fetch_aux()
  File "c:\Users\User\OneDrive\Desktop\FIT3161-3162\FIT3162 Repo10\storage-optimization-in-automated-fulfilment-centers\program\UI.py", line 793, in fetch_aux
    return super().fetch_aux()
           ^^^^^^^^^^^^^^^^^^^
  File "c:\Users\User\OneDrive\Desktop\FIT3161-3162\FIT3162 Repo10\storage-optimization-in-automated-fulfilment-centers\program\UI.py", line 639, in fetch_aux
    raise Exception(self.ex_not_int)
Exception: Input cannot be negative integers, or other types like string
```

Storage Optimization in Automated Fulfilment Centers

Chosen Algorithm: Option 1 - Back Bottom Left Fill

Please Input Box Details (Positive Numbers):

Number of Boxes: -100

Input Number Ranges (Low to High)

Quantity Range	-50	to	-100
Dimensions Range	OraOraOraOra	to	RoadRoller
Weight Range	efjaeoifheiogog	to	ryeyhaeshsht

Allow Varying Priority Levels	Yes
Allow Varying Loading Orientations	Yes
Allow Upside Down Loading	Disabled

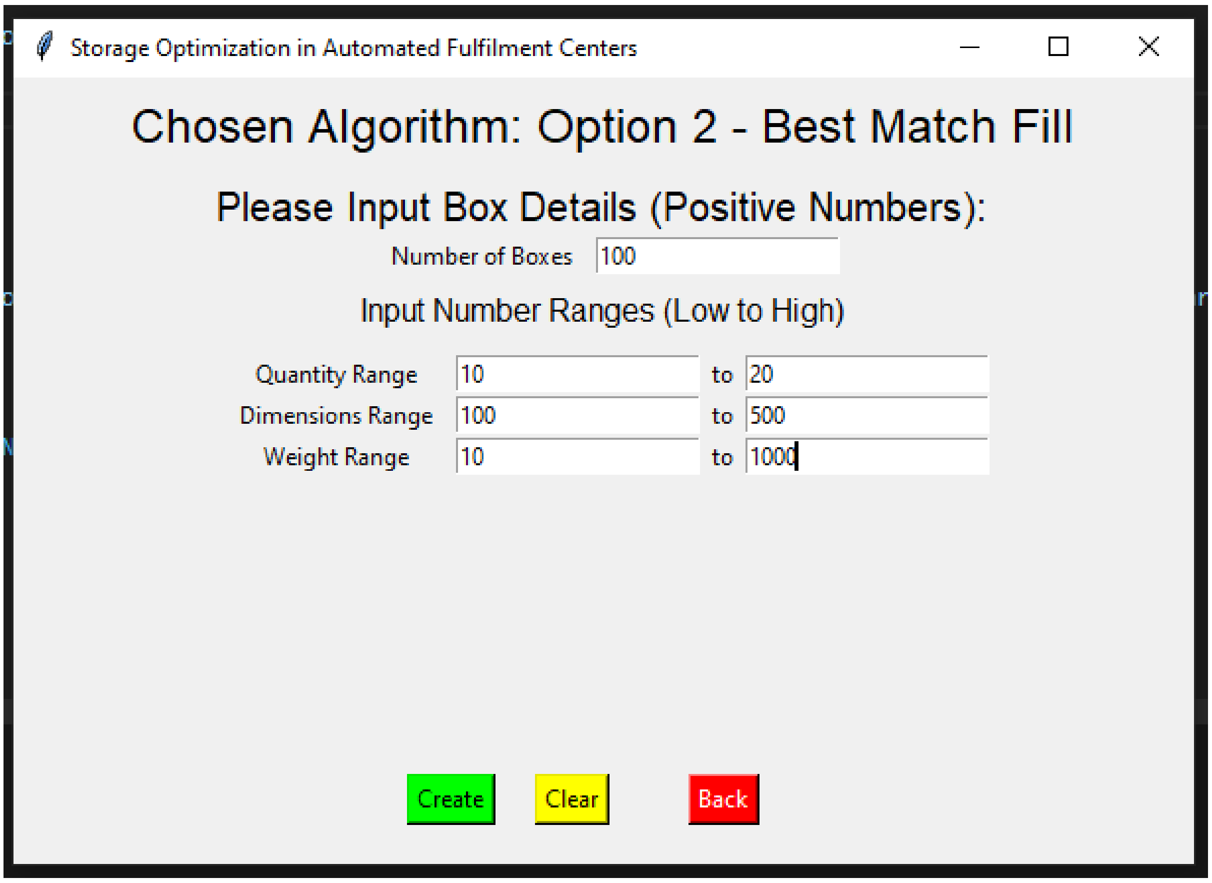
Input cannot be negative integers, or other types like string

Note:

- Since all inputs are positive integers, all the forms are programmed to prevent invalid input from the user (alphabet input, null inputs, etc.)
- Invalid Inputs will throw an exception to the console, and the UI window. This is true for Bin and Box CSV Creation Windows.

2.2.2 Create new box (SKU) CSV file

Upon selecting “Create New Box (SKU) CSV File”, you may be prompted with one of the two different forms depending on which solution algorithm you selected in the main menu.


Storage Optimization in Automated Fulfilment Centers

Chosen Algorithm: Option 2 - Best Match Fill

Please Input Box Details (Positive Numbers):

Number of Boxes: 100

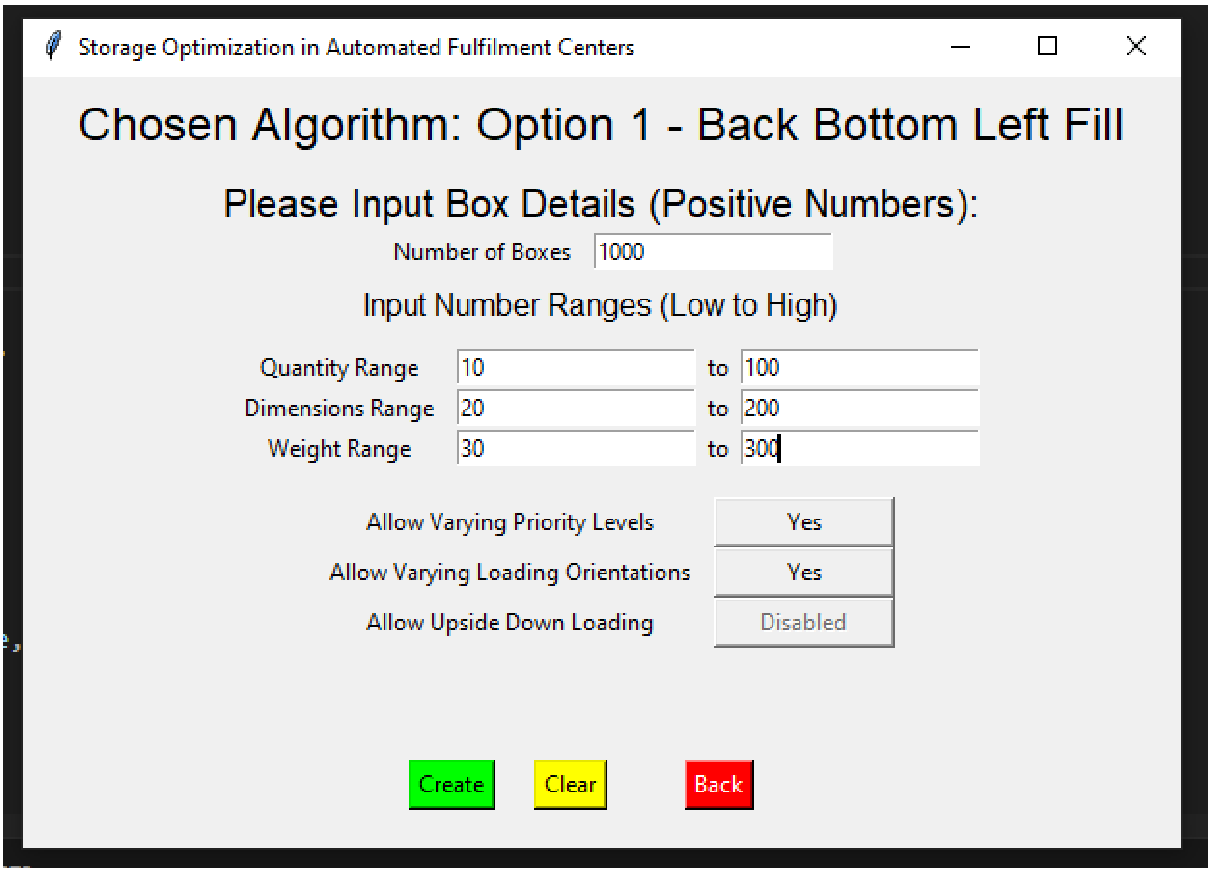
Input Number Ranges (Low to High)

Quantity Range	10	to	20
Dimensions Range	100	to	500
Weight Range	10	to	1000

If your Solution Algorithm choice is Option 2, you may fill in the number of boxes, quantity range, dimensions range, and weight range for generating the file of boxes. In this example, we will generate 100 boxes such that:

- Quantity Range - There can be 10 to 20 types of boxes amongst the 100 boxes, based on their other parameters (dimension, range, etc.)
- Dimensions Range - The Width, Depth, and Height are values within 100 to 500.
- Weight Range - Boxes can have weights ranging from 10 to 1000

When done, select “Create” to generate the Box CSV file. All CSV files will be automatically named and numbered. “Clear” will clear all input fields, and “Back” will return back to the Master Subroutine Window.


Storage Optimization in Automated Fulfilment Centers

Chosen Algorithm: Option 1 - Back Bottom Left Fill

Please Input Box Details (Positive Numbers):

Number of Boxes: 1000

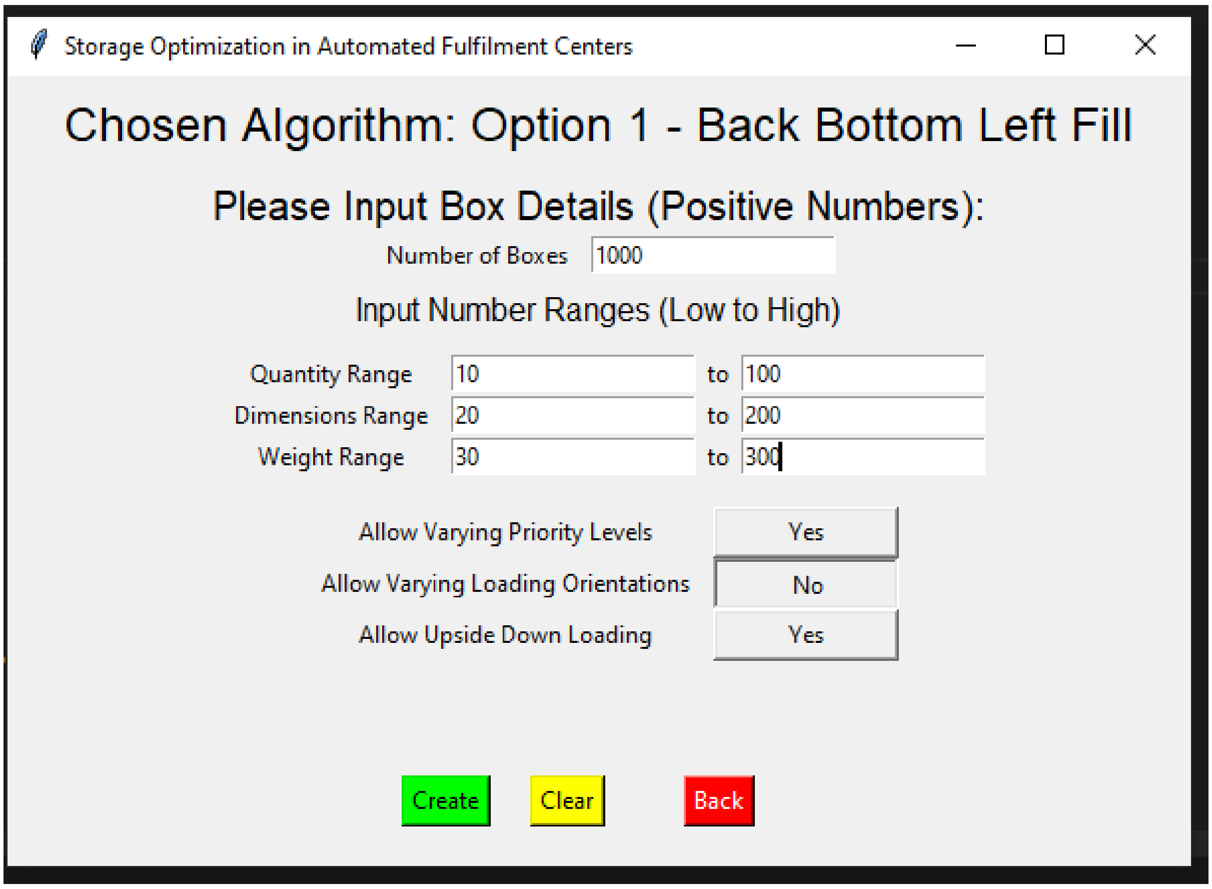
Input Number Ranges (Low to High)

Quantity Range	10	to	100
Dimensions Range	20	to	200
Weight Range	30	to	300

Allow Varying Priority Levels	Yes
Allow Varying Loading Orientations	Yes
Allow Upside Down Loading	Disabled

If your Solution Algorithm choice is Option 1, there are a few additional boolean parameters (Yes - Allow / No - Do not allow) to set due to the difference in algorithm heuristic:

- Allow Varying Priority Levels - Toggles the algorithm's ability to prioritise certain boxes, which helps determine the insert order for nearly identical boxes.
- Allow Varying Loading Orientations - Toggles the algorithm ability to load boxes with different orientations (facing up, down, etc.).


Storage Optimization in Automated Fulfilment Centers

Chosen Algorithm: Option 1 - Back Bottom Left Fill

Please Input Box Details (Positive Numbers):

Number of Boxes: 1000

Input Number Ranges (Low to High)

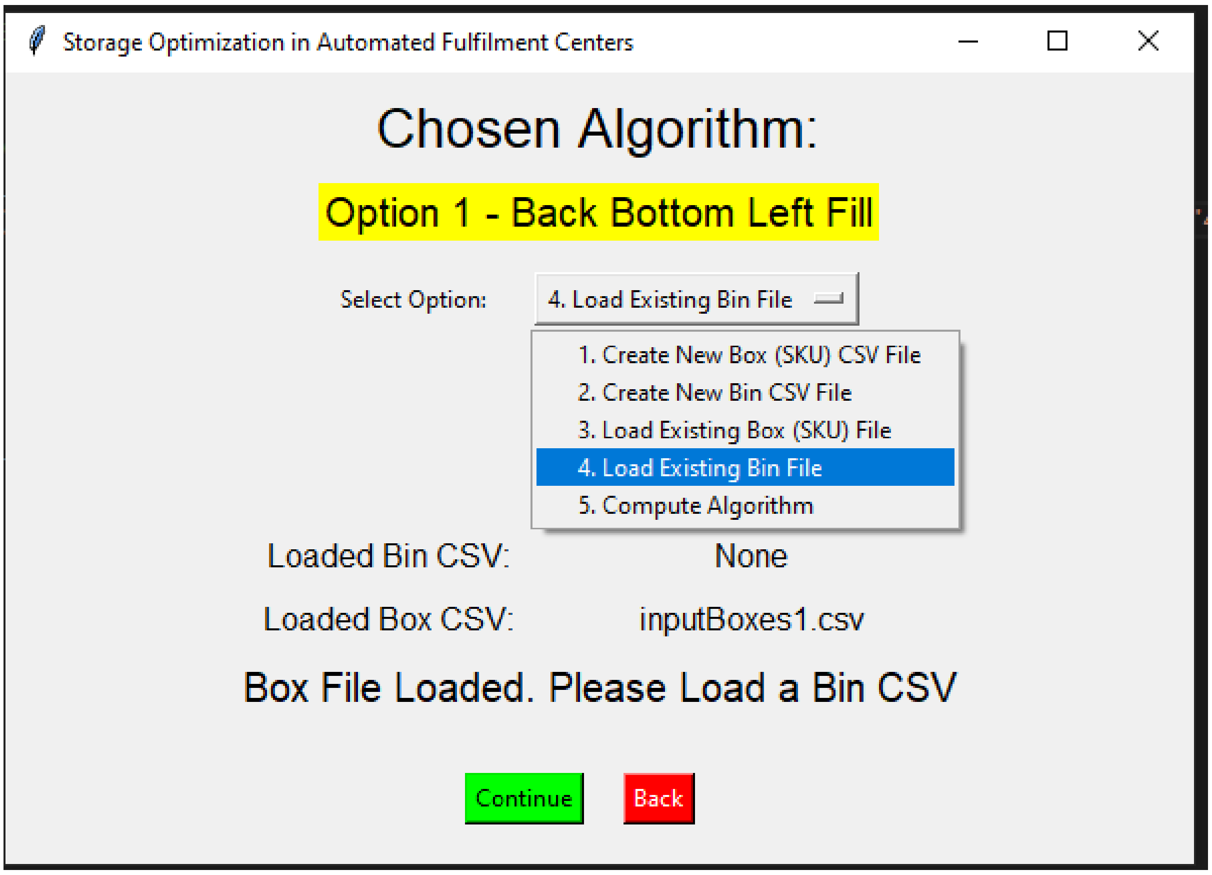
Quantity Range	10	to	100
Dimensions Range	20	to	200
Weight Range	30	to	300

Allow Varying Priority Levels	Yes
Allow Varying Loading Orientations	No
Allow Upside Down Loading	Yes

- Allow Upside Down Loading - If “Allow Varying Loading Orientations” is “Yes”, this is not necessary and is fixed on “No”. Otherwise, this toggles whether boxes can be loaded upside down, or not.

2.2.3 Load existing bin CSV file or box (SKU) file

Once the Bin and Box CSV files are created, they will be automatically placed in their respective file paths for the selected solution algorithm.


Storage Optimization in Automated Fulfilment Centers
Select Option:
4. Load Existing Bin File
1. Create New Box (SKU) CSV File
2. Create New Bin CSV File
3. Load Existing Box (SKU) File
4. Load Existing Bin File
5. Compute Algorithm
Continue
Back

Chosen Algorithm:

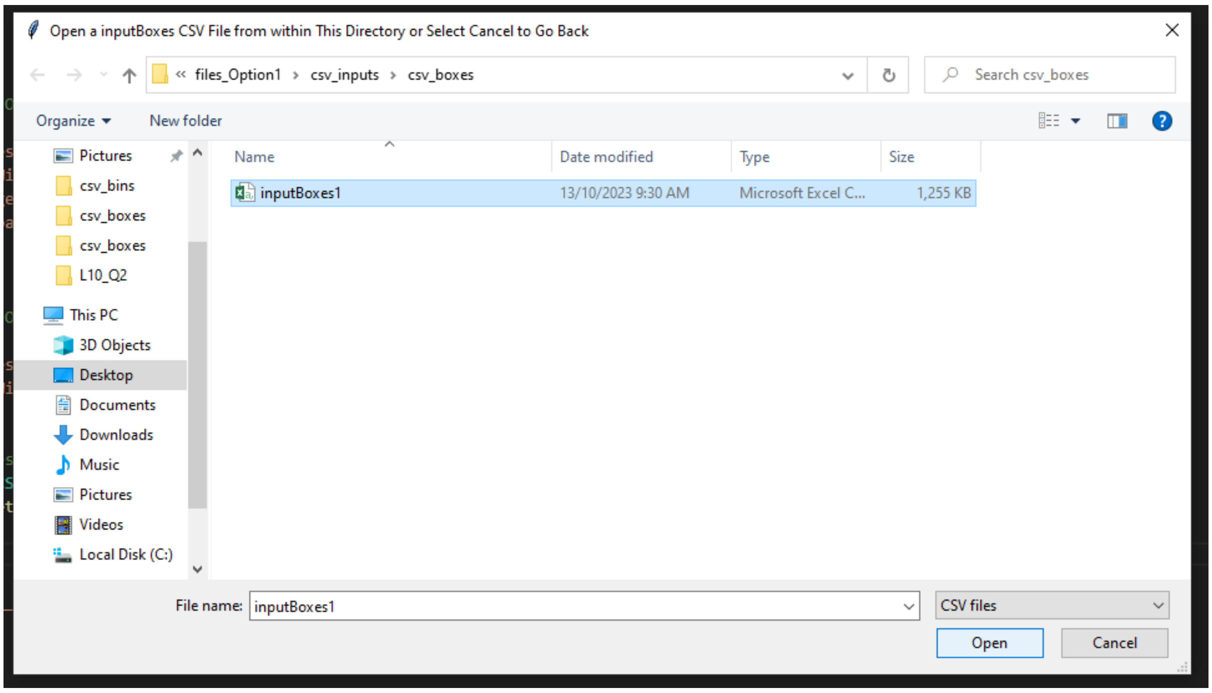
Option 1 - Back Bottom Left Fill

Loaded Bin CSV: None

Loaded Box CSV: inputBoxes1.csv

Box File Loaded. Please Load a Bin CSV

Selecting “Load Existing Bin CSV File” and “Load Existing Box CSV File” will open a file explorer to their respective directories.


Open a inputBoxes CSV File from within This Directory or Select Cancel to Go Back

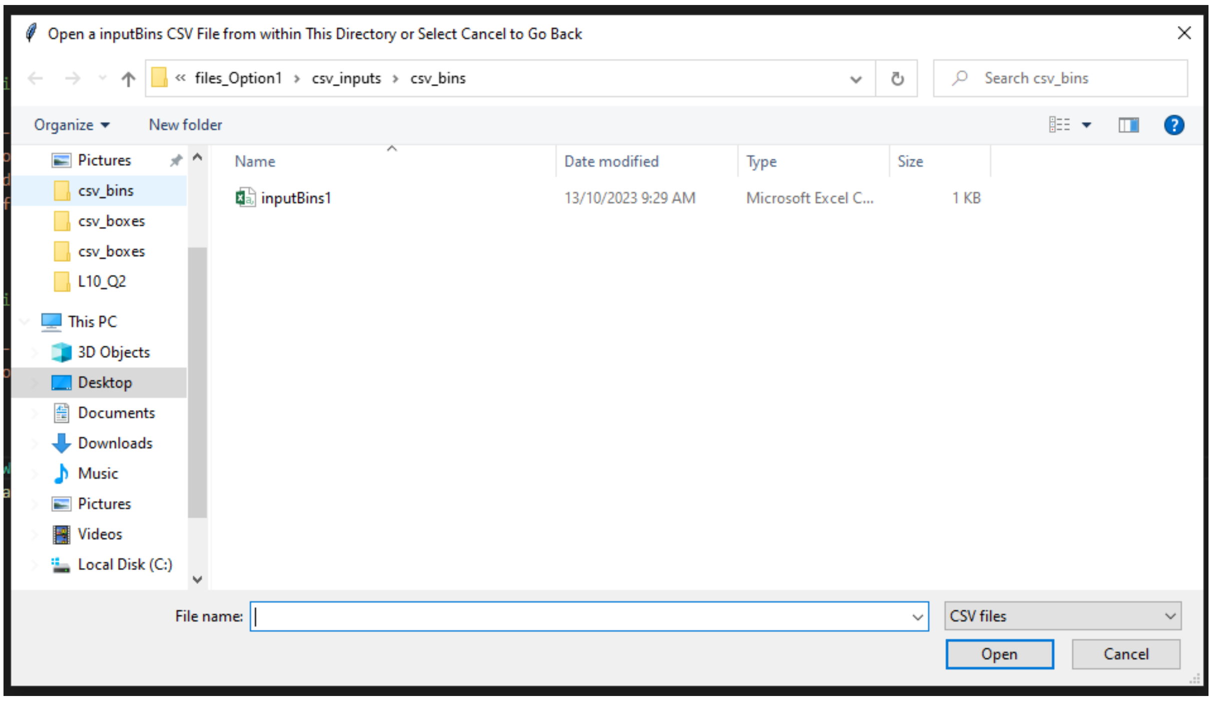
« files_Option1 › csv_inputs › csv_boxes

Name	Date modified	Type	Size
inputBoxes1	13/10/2023 9:30 AM	Microsoft Excel C...	1,255 KB

File name: inputBoxes1

CSV files

To load the Box CSV file, just select the file, then select “Open”. To go back, select “Cancel”.


Open a inputBins CSV File from within This Directory or Select Cancel to Go Back

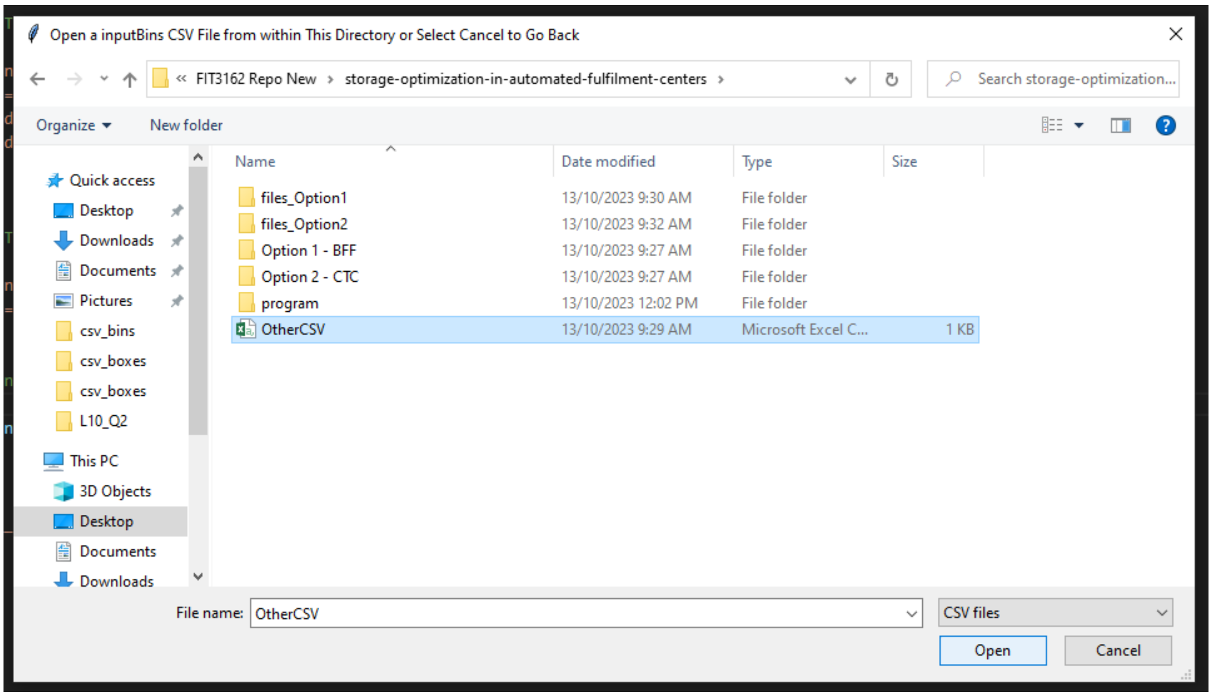
« files_Option1 › csv_inputs › csv_bins

Name	Date modified	Type	Size
inputBins1	13/10/2023 9:29 AM	Microsoft Excel C...	1 KB

File name:

CSV files

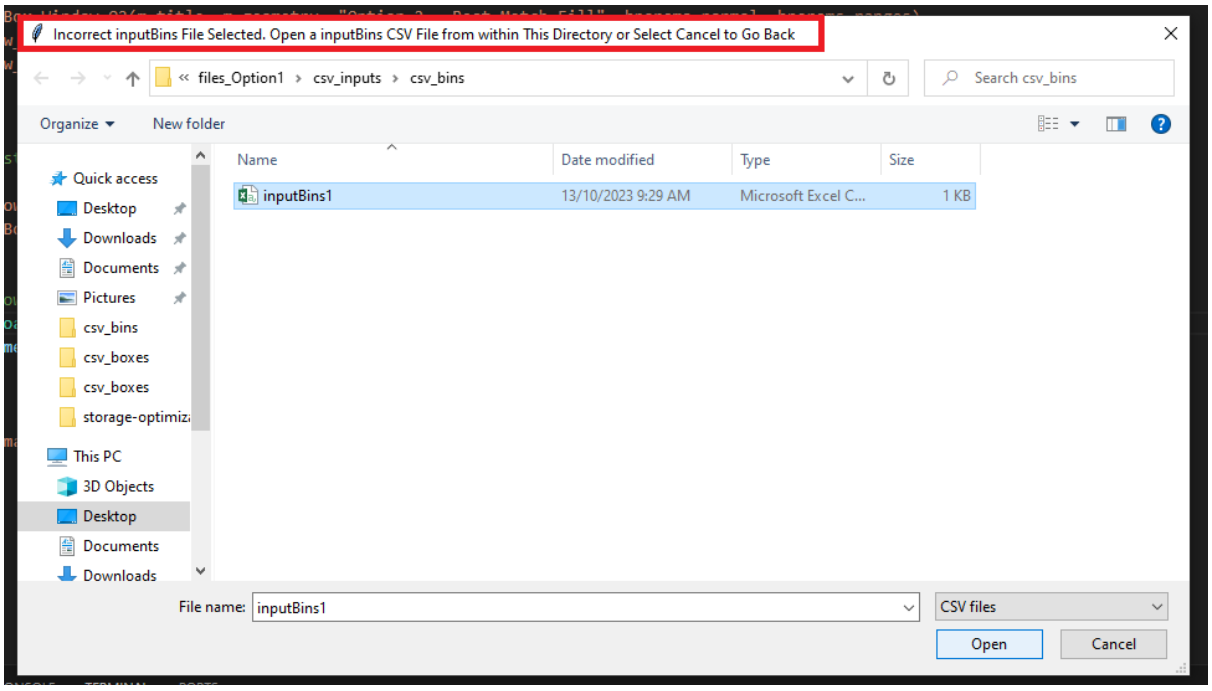
The same applies to Bin CSV files.


Open a inputBins CSV File from within This Directory or Select Cancel to Go Back

« FIT3162 Repo New › storage-optimization-in-automated-fulfilment-centers ›

Name	Date modified	Type	Size
files_Option1	13/10/2023 9:30 AM	File folder	
files_Option2	13/10/2023 9:32 AM	File folder	
Option 1 - BFF	13/10/2023 9:27 AM	File folder	
Option 2 - CTC	13/10/2023 9:27 AM	File folder	
program	13/10/2023 12:02 PM	File folder	
OtherCSV	13/10/2023 9:29 AM	Microsoft Excel C...	1 KB

File name: OtherCSV
CSV files


Incorrect inputBins File Selected. Open a inputBins CSV File from within This Directory or Select Cancel to Go Back

« files_Option1 › csv_inputs › csv_bins

Name	Date modified	Type	Size
inputBins1	13/10/2023 9:29 AM	Microsoft Excel C...	1 KB

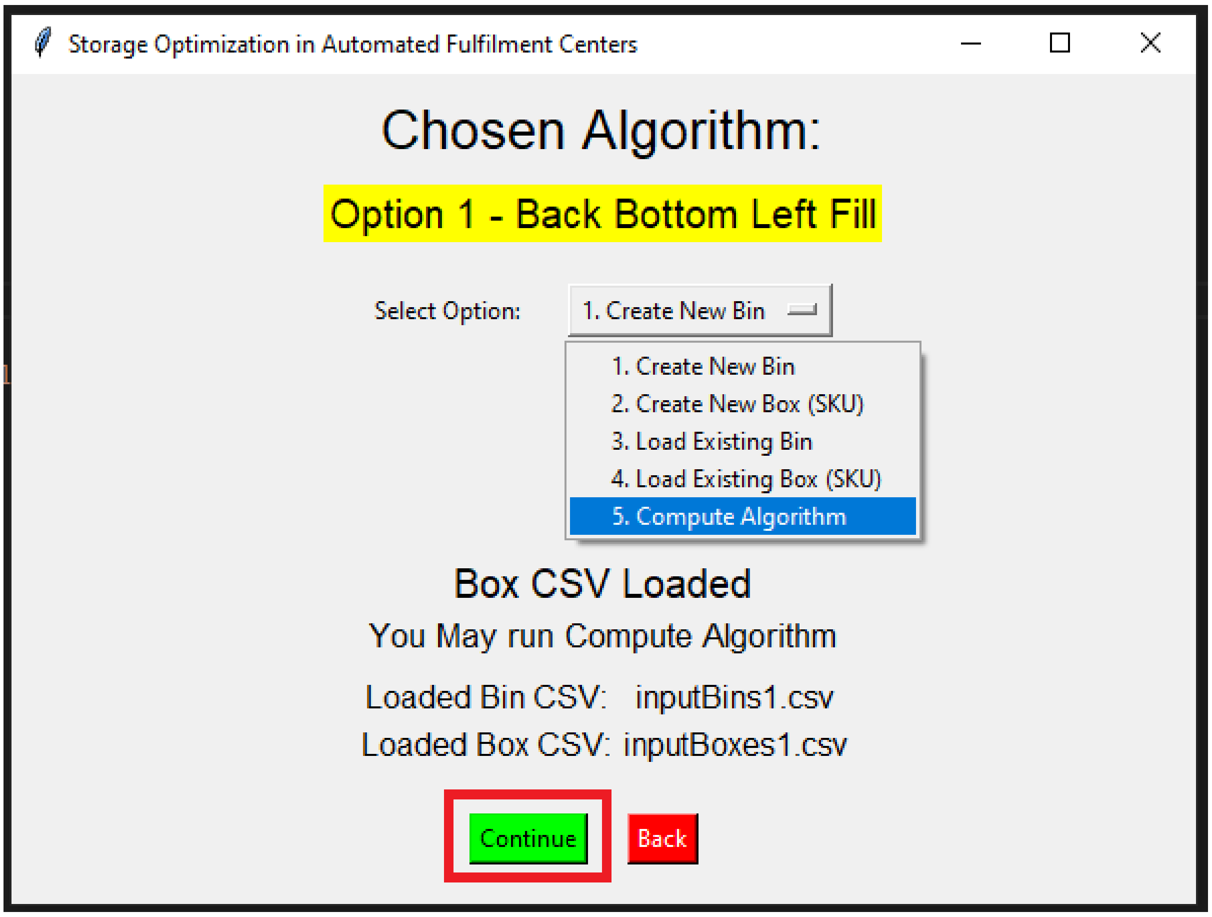
File name: inputBins1
CSV files

Note:

- If the user navigates to other folders, and selects other csv files, the UI will navigate back to the original directory, and display a message to notify that an incorrect file was selected.
- Other file types (excel, word, etc.) cannot be loaded into the program.

2.2.4 Execute packing algorithm

Once both an input bin and input box CSV file have been loaded, you will be permitted to execute the packing algorithm. The UI will indicate when you may do so once both a box and a bin CSV file are loaded, and the option to compute will appear in the dropdown menu. Select the *"Compute Algorithm"* option to do so.



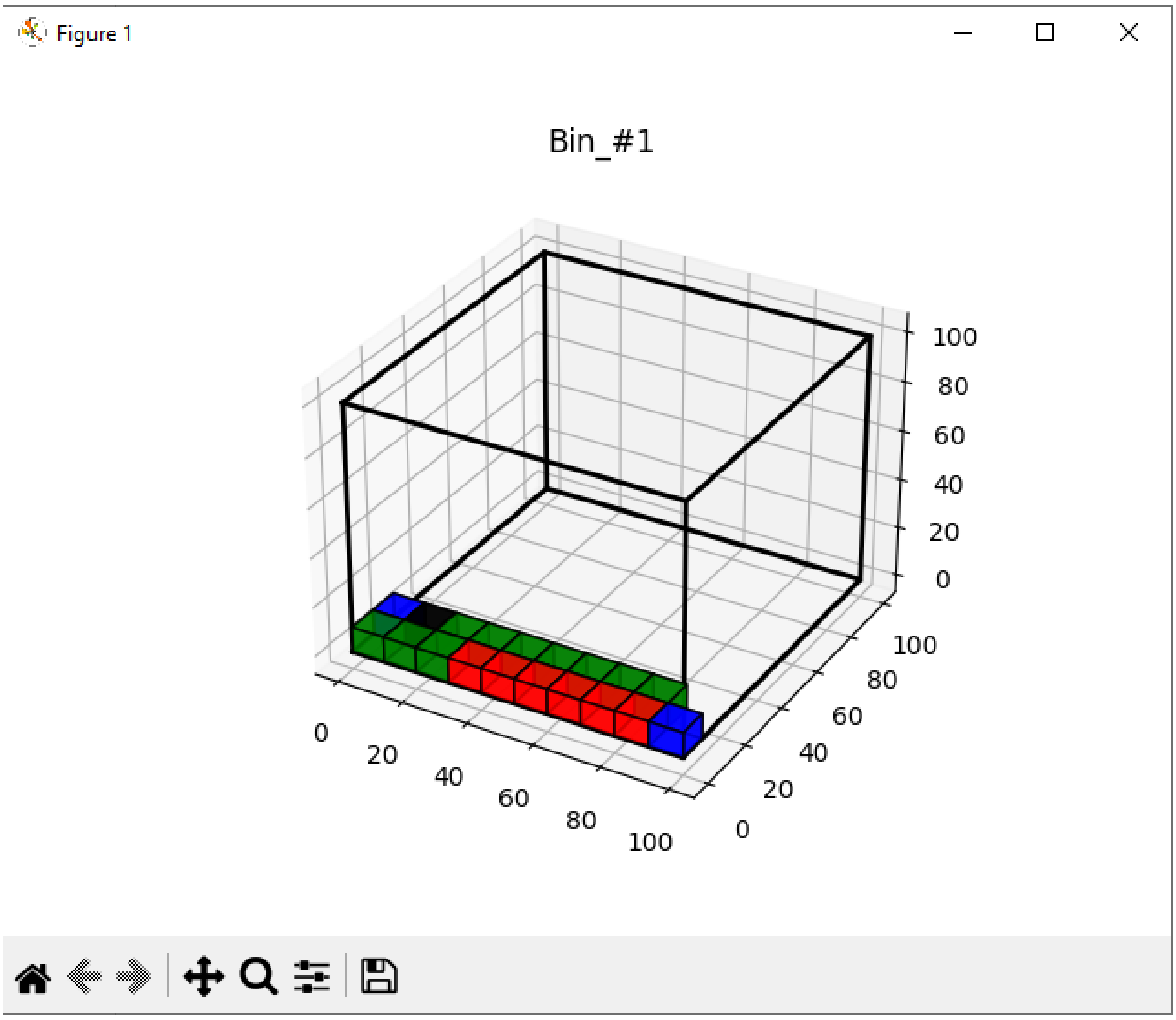
Box CSV Loaded

You May run Compute Algorithm

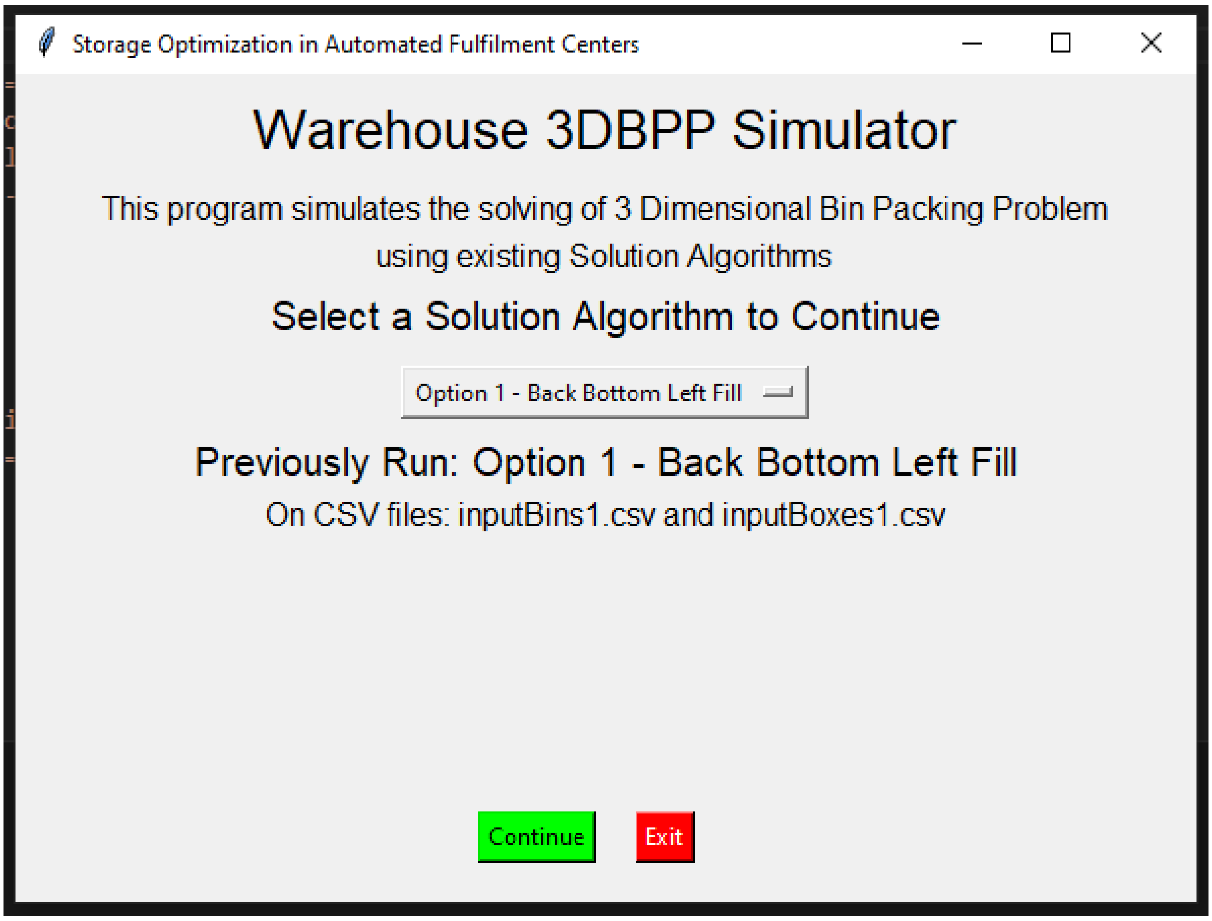
Loaded Bin CSV: inputBins1.csv

Loaded Box CSV: inputBoxes1.csv

The computation may take a while depending on the volume of data the algorithm is handling and the speed of your device. When the computation has finished, several windows will open displaying the contents of each bin, as follows.



The program does not proceed until all windows have been closed. Once that is done, the relevant output files will be generated in their corresponding directory.


Storage Optimization in Automated Fulfilment Centers

Warehouse 3DBPP Simulator

This program simulates the solving of 3 Dimensional Bin Packing Problem using existing Solution Algorithms

Select a Solution Algorithm to Continue

Previously Run: Option 1 - Back Bottom Left Fill

On CSV files: inputBins1.csv and inputBoxes1.csv

Once the computation is done and all bin 3D windows are closed, the program will return to the main menu, which will contain the details for the previously run algorithm and files. At this point, you may choose to run another algorithm, or select “Exit” to leave the program.

For now, that is all. Thank you for using our 3DBPP simulation program, and we do hope the experience was good for you. Do let us know if there are any rooms for improvement.