Installing coincnbc

conda install -c conda-forge coincbc

* Didn’t work.
* Came across this <https://stackoverflow.com/questions/58868054/how-to-install-coincbc-using-conda-in-windows> with the following answers being helpful
  + <https://stackoverflow.com/a/75160274>
  + <https://stackoverflow.com/a/66178957>
    - These led me to download the zip file Cbc-master-win64-msvc16-mdd.zip which is highlighted in the img below:Graphical user interface, text, application

      Description automatically generated



* + - I unzipped the file
    - I copied the contents of the bin directory (Cbc-master-win64-msvc16-md\bin)
      * Note that I replaced any files where windows asked me if I wanted to
    - I pasted these .exe and .dll files in anaconda\envs\my\_env\Library\bin
    - I think that is all?

After installing it I realized that this may work:

$ conda install coin-or-cbc

So should test that if you get a chance

* PackagesNotFoundError: The following packages are not available from current channels:
* coin-or-cbc

So it looks like coin cbc will replace glpsol in line 205 of main.py. Trying to work out how it would be implemented:

<https://www.gams.com/archives/presentations/present_gamslinks.pdf>

is this esstnially what we want? <https://manpages.ubuntu.com/manpages/xenial/man1/clp.1.html>

* <https://github.com/coin-or/Clp>
* Note that for windows in download, the readme says: Windows: The easiest way to get Clp on Windows is to download an archive as described above.
* Also:
  + <https://forum.openmod.org/t/osemosys-with-coin-or-clp-solver/190>

**maybe cbc is actually what we want:**

<https://groups.google.com/g/osemosys/c/QCDeNevGzdo?pli=1>

and this <https://github.com/OSeMOSYS/otoole/blob/master/docs/functionality.rst> which specifies how to use otoole results with cbc, there is a more detailed description here: https://otoole.readthedocs.io/en/latest/functionality.html

Probably want to use otoole results

* I think this shows how to create the .lp file I think we need: https://osemosys.readthedocs.io/en/latest/manual/Advanced%20functionalities.html?highlight=glpsol#using-osemosys-with-the-solver-cplex

3/2/2023

1. Updating otoole to v 1.0.0 from v 0.11.0:

Pip install otoole -U

To reinstall v0.11 then use pip install --force-reinstall -v "otoole ==0.11.0"

1. Creating a new data config.yaml where the short argument is change to short\_name because it seems the new version requris this
   1. This is because the read and write functions of otoole require the short\_name argument, rather than ‘short’. Also there were flow on effects required:
      1. Created a data\_config\_short\_names because we need to be able to compare the dataconfig to the excel file outside of the otoole function, this is essentially what the otoole functiuon does anyway.
2. Created a new results config and set all type: args to result instead of var, and also included the two fields in every entry:
3. default: 0
4. calculated: "True"

Could be that default could be 1, and calculated could be False, I just assumed not, cause it was easy

* I also created E:\APERC\power-model\src\results\_config\_copy\_test.yml which has the changes above and removed ,Value from the indices argument for every entry

1. Just a thought, It would be good to consider how to programmatically create or at least check the config files are as we expect. This idea came because I removed the Fuel parameter from the EmissionActivityRatio entry in data\_config.yml (the copy version for now)
   1. Perhaps a bunch of testing functions

Issues ive had incase they pop up again:

OtooleConfigFileError:

RateOfActivity -> Index not in user supplied sets

* This occurs because you need to include one of the indices in results or data config as a set