User Manual

SimCCS

v1.0 – January 2018

If using this application as part of published research, please cite the following publication:

> Middleton, R.S., Yaw, S., Wang, Y., Ellett, K., *SimCCS*$^{2.0}$: An open-source tool for optimizing CO_2 capture, transport, and storage infrastructure, In Review.

For assistance, collaboration, or comments, please contact:

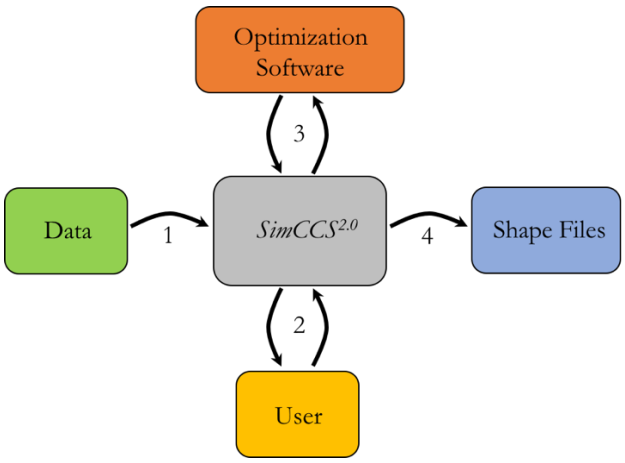
Sean Yaw, yaw@lanl.gov

Richard Middleton, rsm@lanl.gov

Table of Contents

Introduction

Carbon Capture and Storage (CCS) is the process of capturing CO_2 from industrial processes, transporting it, and injecting it into geologic reservoirs for long term storage. *SimCCS* is a software application that enables researchers to design CCS infrastructure networks, such as, for determining which sources and sinks to use, and where to build pipelines to inject a target quantity of CO_2. *SimCCS* uses user-provided data covering source (capture capacities and costs), sink (geologic parameters), and transportation (weighted-cost surface of the deployment region) information. From this user supplied data, *SimCCS* creates candidate transportation network edges and formalizes an optimization problem that will determine the most cost-effective deployment. This optimization problem can be solved using third party software (e.g. CPLEX), and the solutions can be brought back into *SimCCS* for visualization and analysis. *SimCCS* consists of a graphical user interface (GUI) that is used to drive data management, pipeline network generation, and optimization model formulation, solution, and display. This workflow is illustrated below. The goal of this manual is to provide a basic overview of the functionality of *SimCCS*, detail its basic design, outline the data formats, and provide examples of its functionality. Source code should be referenced for details.

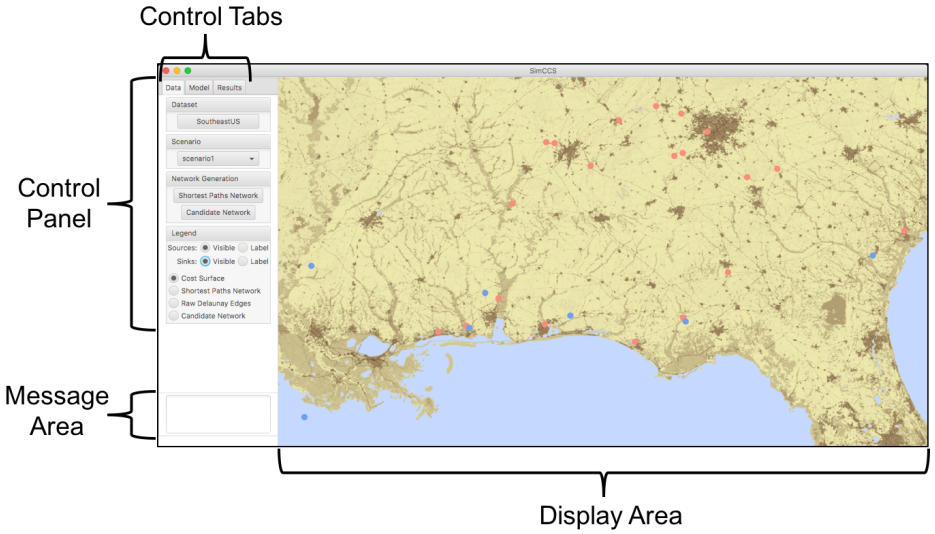


Installation

SimCCS is freely available in a GitHub repository (https://github.com/simccs/SimCCS). *SimCCS* is written in Java 8 and runs on the Java Virtual Machine, thus there is no formal installation procedure. *SimCCS* can either be run from source (compile and execute code) for users looking to modify capability, or by running the jar file in the repository (store/SimCCS.jar) for users seeking a simpler startup procedure. *SimCCS* is packaged will all the required dependencies (openmap for building shape files).

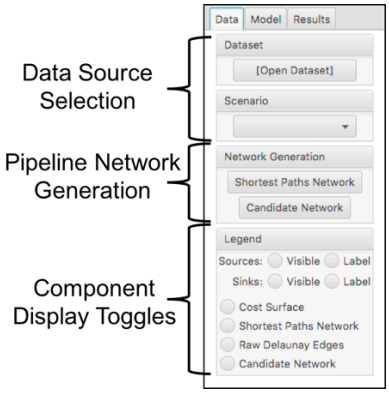
Basic Functionality

The *SimCCS* control panel consists of three different tabs to drive data management, model generation and solving, and solution display. This section will provide an overview of the functionality of each of these tabs.



Data Tab

The data tab enables the user to load and manipulate source/storage/transportation data, generate pipeline networks, and determine what data components are displayed. Data formats are described later in this manual.

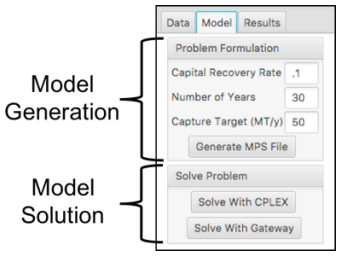


- *[Open Dataset]* button. Direct *SimCCS* to the dataset source.
- *Scenario* selector. Select specific scenario in the data source. The options are automatically populated when the dataset is selected.
- *Shortest Paths Network* button. Generate and save the all-pairs shortest paths between all source/storage locations. This button forces the network to be regenerated and saved, overriding any saved versions.

- *Candidate Network* button. Generate and save the edge-reduced candidate pipeline network. This button forces the candidate network to be regenerated and saved, overriding any saved versions. Current algorithm is the Delaunay triangulation algorithm, but can be modified.
- Legend toggle buttons. Toggle display of various data components.

Model Tab

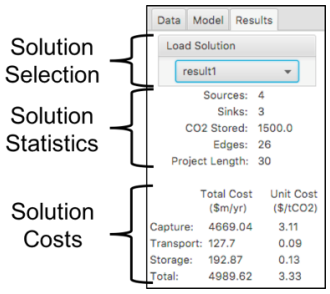
The model tab allows users to formulate and solve optimization models in the form of mixed integer linear programs (MILP) representing the optimal deployment of CCS infrastructure that satisfies the system constraints (namely source/storage/transportation costs and target capture amounts). The user is able to build the model into an MPS file and then has two options for solving it.



- *Capital Recovery Rate*. Percent of fixed costs paid per year.
- *Number of Years*. Number of years in project.
- *Capture Target*. Amount of CO_2 (MT) targeted for capture per year.
- *Generate MPS File* button. Build the MPS file.
- *Solve with CPLEX* button. Solve MPS file locally with CPLEX. If CPLEX is not installed or not on the System path, an error is displayed.
- *Solve with Gateway* button. Solve the MPS offsite with Indiana University HPC resources. Requires login and navigation. See Science Gateway section for details.

Results Tab

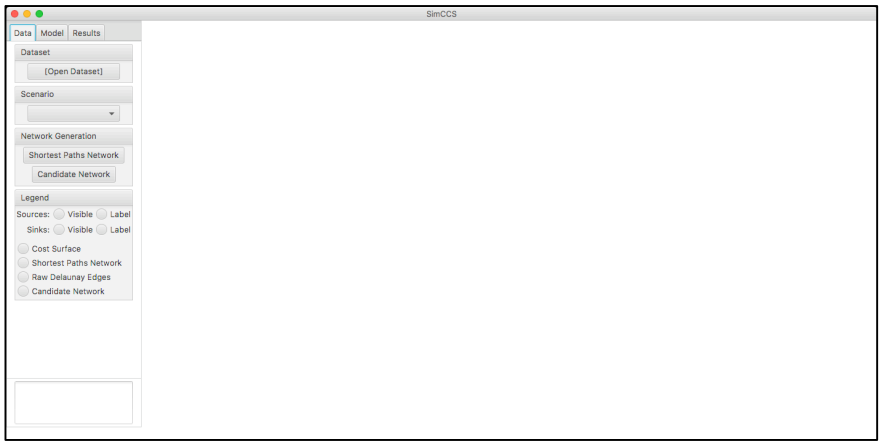
The results tab allows users to visualize solutions and create ShapeFiles.



- Solution Selection. Select solution from set of available solutions.
- Solution Statistics. Displays solution instance statistics.
- Solution Costs. Displays annualized and unit (tCO_2) costs of given solution.

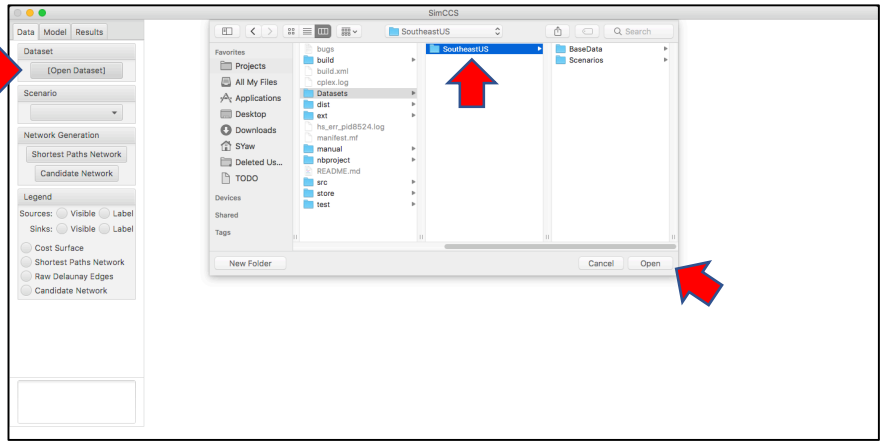
Case Study Example

The *SimCCS* repository includes a sample dataset (Datasets/SoutheastUS) for the purposes of illustrating the required data format (see section below) and providing a step-by-step example. The following guide will walk through all of *SimCCS*'s core functionality using this dataset. To begin, start *SimCCS* by running the jar (double click store/SimCCS.jar) or compiling and executing the source code. This should open up the main GUI.

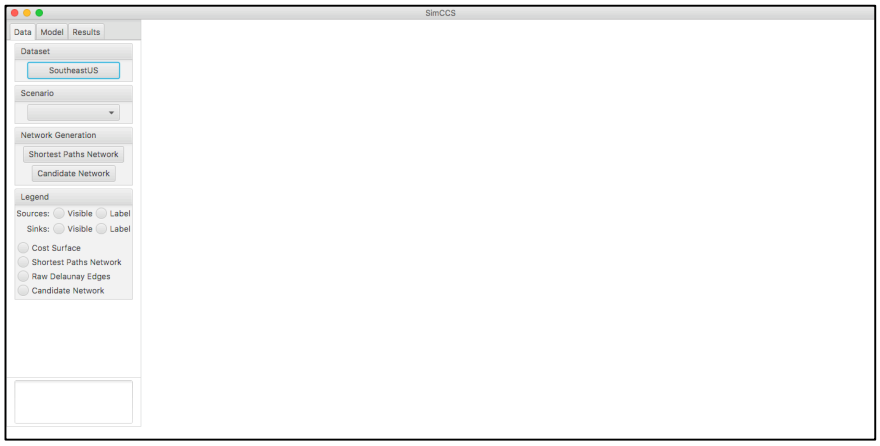


Load Data

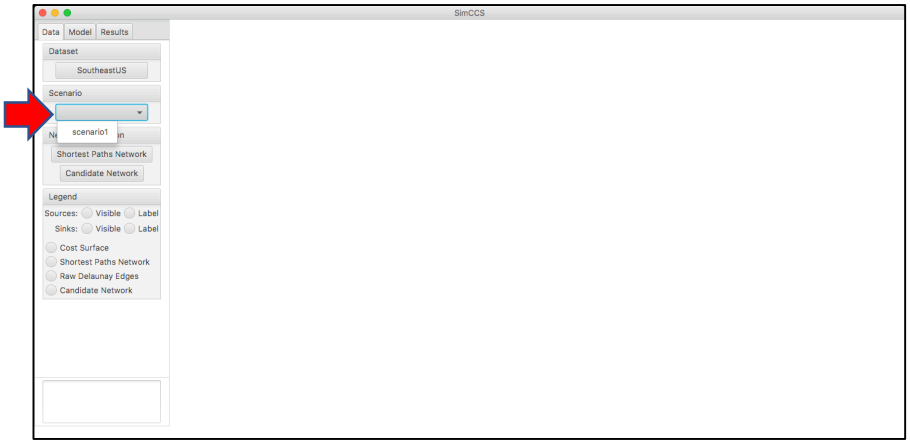
Direct *SimCCS* to the dataset by selecting *[Open Dataset]* and navigating to the *SoutheastUS* folder in the *Datasets* folder. Select the *SoutheastUS* folder.



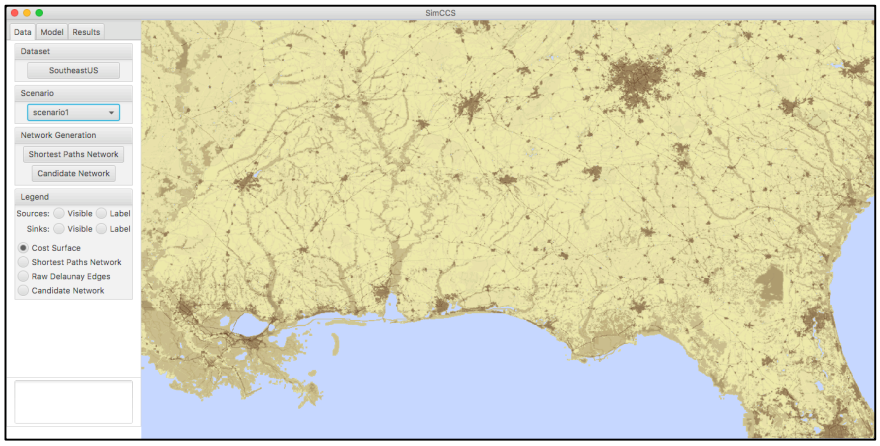
The *[Open Dataset]* text is replaced with *SoutheastUS* to indicate which dataset is currently loaded.



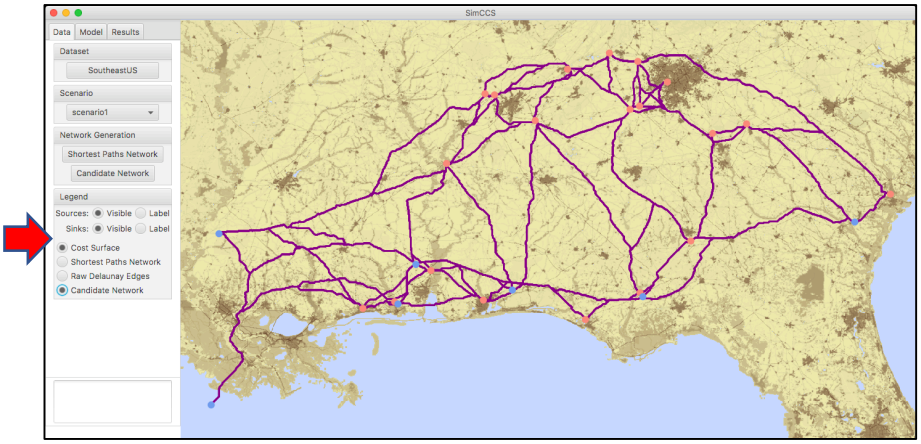
A specific scenario must now be selected. Many scenarios can share the same base cost surface, so multiple scenarios can be contained in a single dataset. Select the dropdown *Scenario* menu and select *scenario1*, the only scenario included in this dataset.



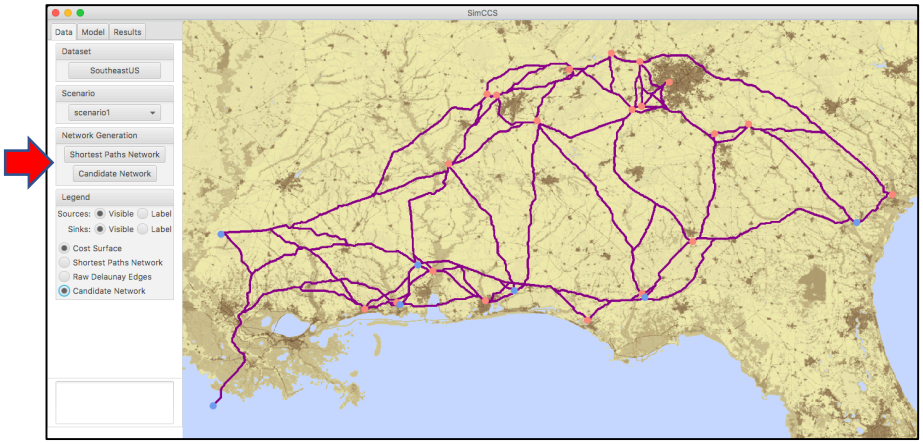
Once a scenario is selected, *SimCCS* will display the cost surface for that dataset and load the source, storage, and network data in the scenario.



Toggle display components by selecting and deselecting buttons in the *Legend* region. This will show the sources, sinks, and any available networks. The display map can also be zoomed with the mouse scroll wheel and moved with click and drag motions to explore more detailed areas.

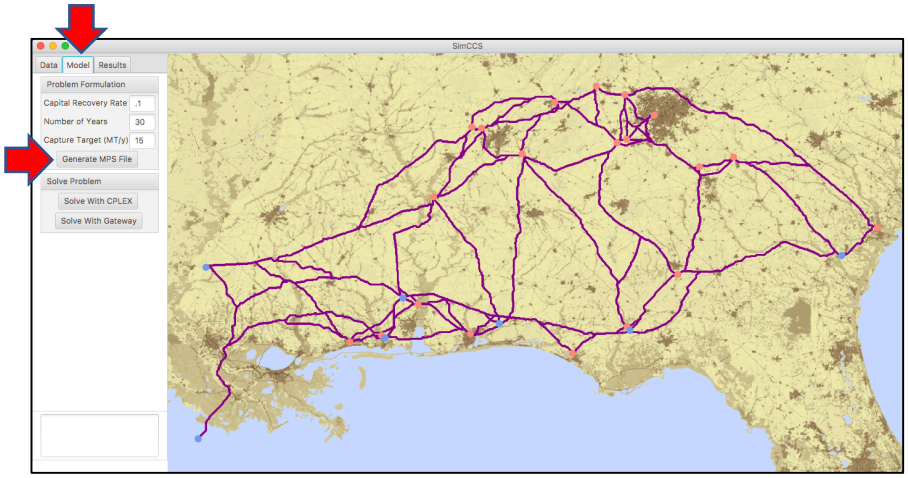


The raw shortest paths network and candidate network can be forced to be regenerated by selecting the applicable buttons in the *Network Generation* region. They are also automatically generated if they do not exist. These buttons are used frequently when testing new network generation algorithms.

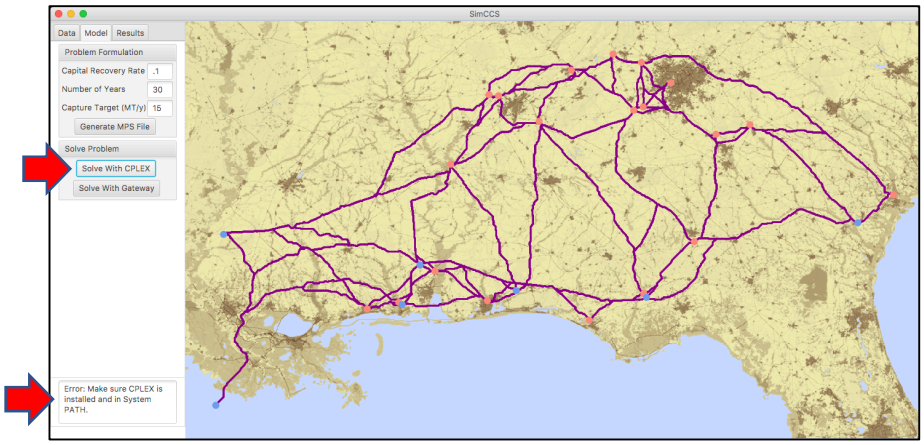


Build/Solve Optimization Model

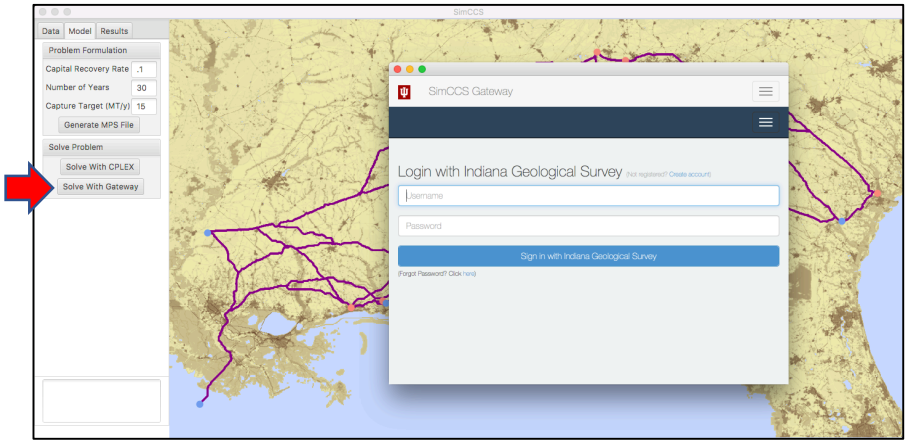
Once data is loaded, *SimCCS* can be used to build and solve CCS design optimization problems by going to the *Model* control tab. Modify the *Capital Recovery Rate*, *Number of Years* (project length), and *Capture Target* fields to the desired values and select *Generate MPS File* to build the optimization model.



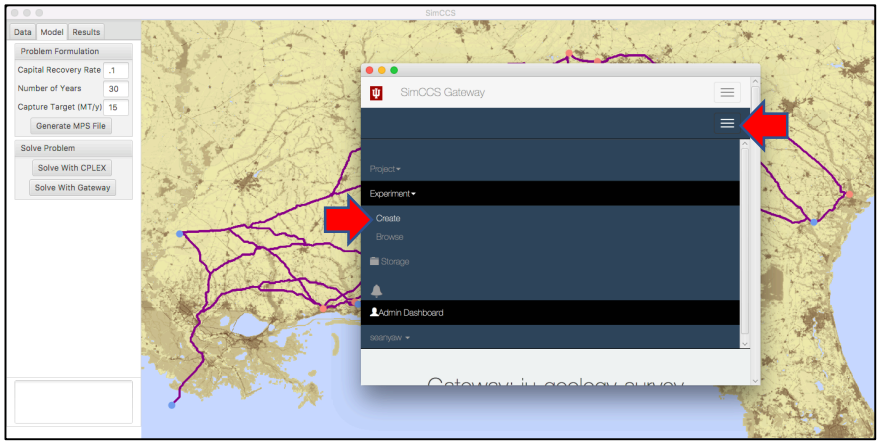
There are three options for solving the model (MPS file). The first is to use a solver external to *SimCCS*. This is done by directly solving the MPS file and putting the solution file into the correct location (see Data Format and Organization section). The second is to use IBM CPLEX that is already installed on the host machine. To use this option, simply select *Solve With CPLEX*. *SimCCS* handles all of the file organization. If CPLEX is not installed, or installed, but not in the system path, an error will be displayed.



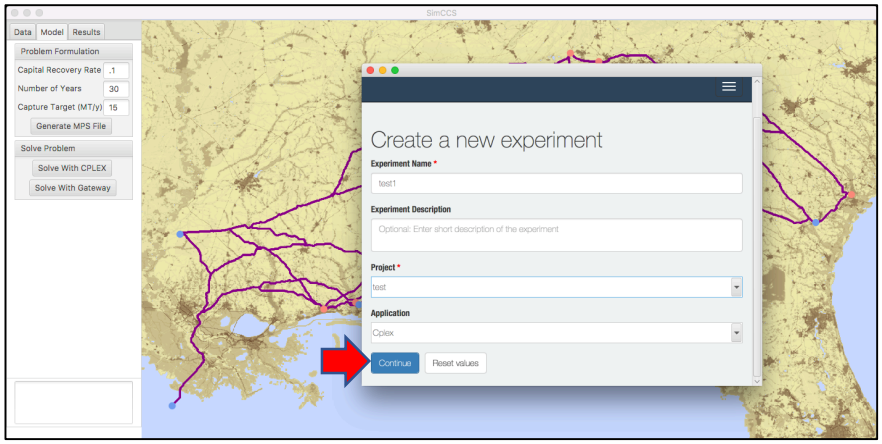
The third option for solving MPS files is to use Indiana University's SimCCS Science Gateway. To begin, select *Solve With Gateway* to launch the web interface. This service requires an account, so initially, select the *Create account* option and afterwards login with credentials.



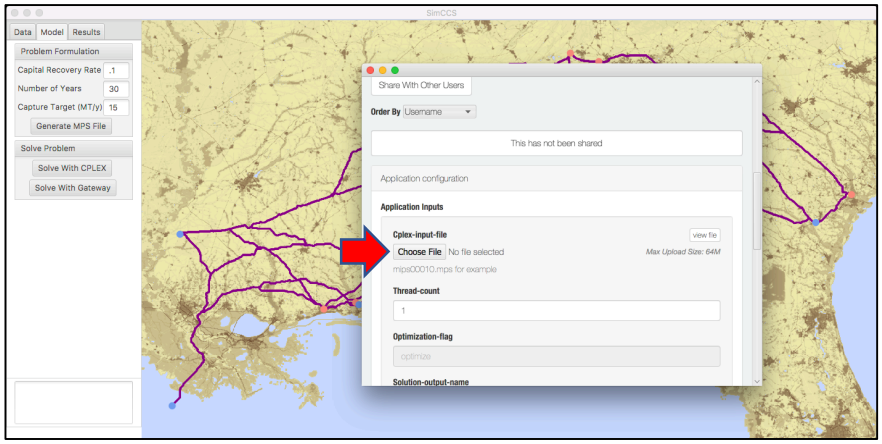
Once logged in, select the second-tier menu and navigate to *Experiment* -> *Create*.



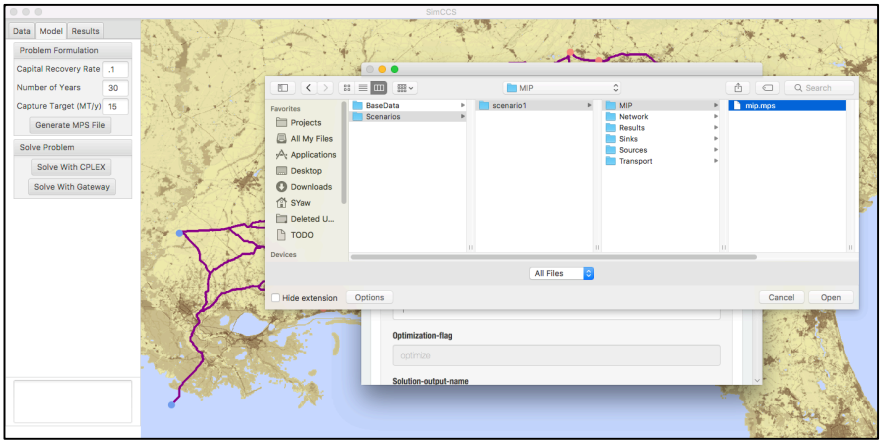
On the experiment creation page, input a name for the experiment and select *Continue*.



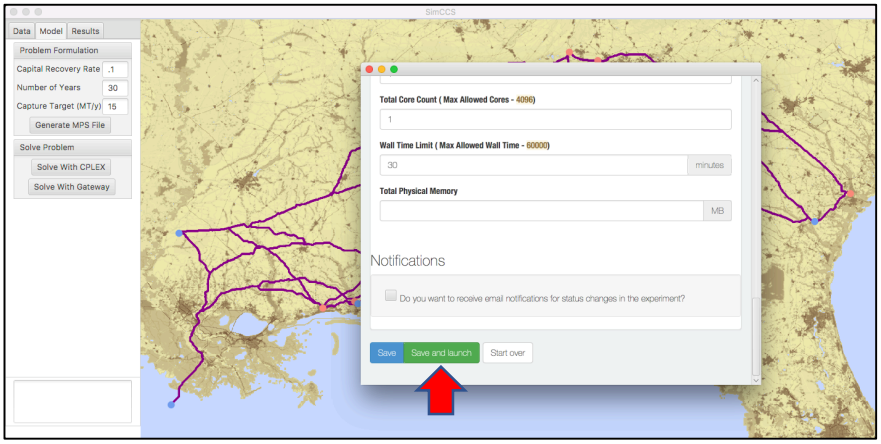
Scroll down on the next page and select *Choose File* to direct the Gateway to the MPS file.



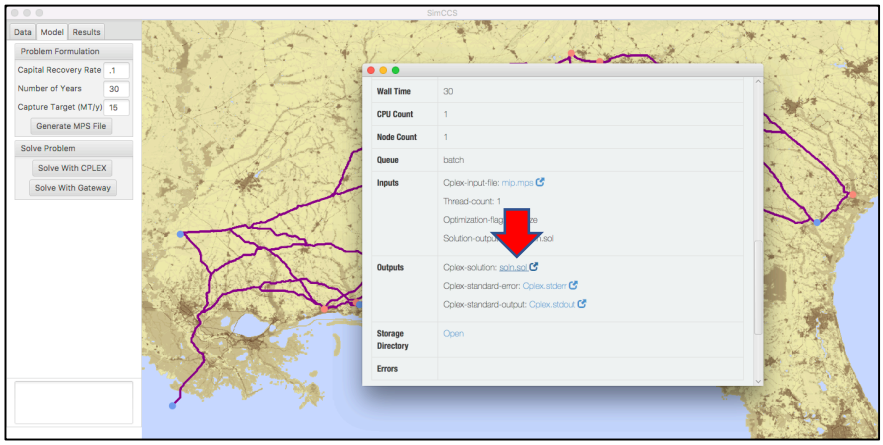
In the popup window, navigate to *SoutheastUS* -> *Scenarios* -> *scenario1* -> *MIP* -> *mip.mps* and select *Open*.



Complete experiment creation by selecting *Save and launch*.

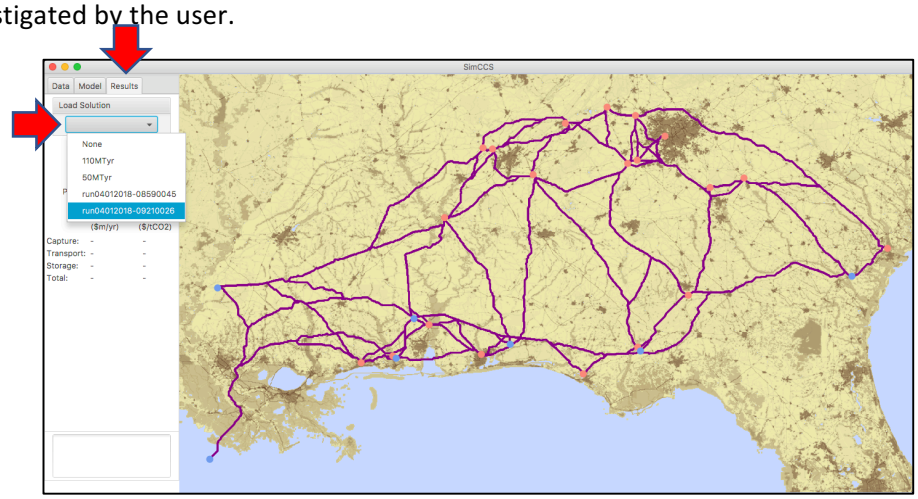


The experiment status page comes up and gives the status of the MPS solution process. When the process completes, scroll down and select *soln.sol*. This will automatically download the solution file and organize the files into the appropriate format. Note: *SimCCS* does not need to remain open while the Gateway is processing. If the Gateway window is closed, relaunch the Gateway (*Solve With Gateway*) and navigate to *Experiment -> Browse*, find the experiment that was created and select *soln.sol*.

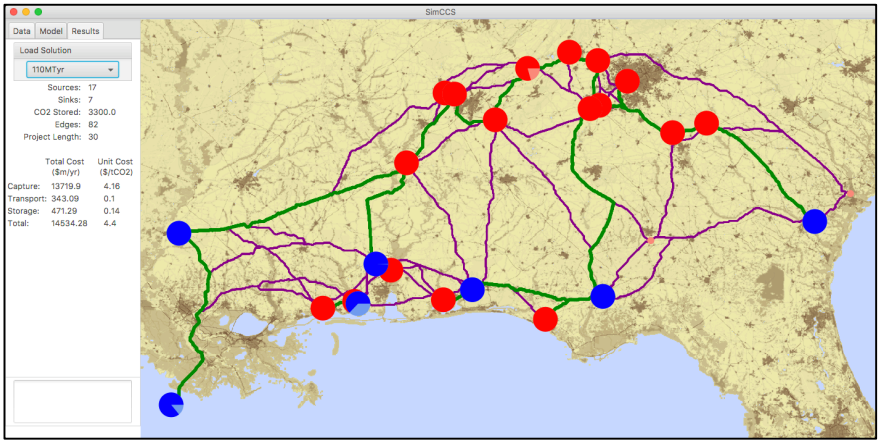


View Results

After a model is build and solved, the results can be viewed in the *Results* control tab. The SoutheastUS dataset comes preloaded with two solutions. Begin by selecting the Results control tab and selecting the Load Solution menu to see available solutions. There will be at least two solutions. 110MTyr and 60MTyr are the two solutions that came preloaded. Other options are the result of model solutions instigated by the user.

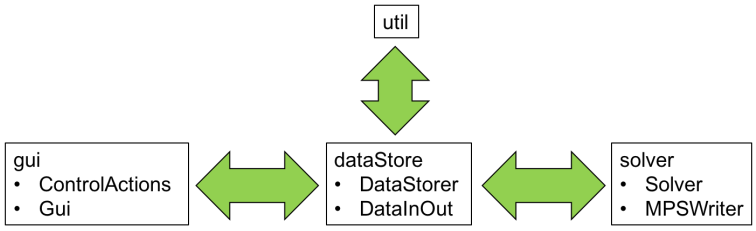


Select a solution to see it in the display area. Solution statistics and costs are populated in the control tab. Underlying source, storage, and network display components can be toggled as before from the *Data* control tab.



Design Details

SimCCS is written in Java 8 and is organized into four key classes tasked with the general goals of data management, GUI management, algorithm library, and utility functions. An illustration of the package structure, key classes, and relationship between them is shown below.



DataStorer

The DataStorer class manages the data for the loaded problem including source, sink, and network data. It also contains the cost surface data and HashMaps representing the graph of the candidate network.

DataInOut

The DataInOut class manages the interface between the actual data files and the DataStorer. It is tasked with reading and writing data and is the location where file format enforcement occurs.

Gui

The Gui class builds the GUI and attaches buttons to actions.

ControlActions

The ControlActions class serves as the interface between GUI buttons and data. It issues data requests to the DataStorer and updated the GUI accordingly.

Solver

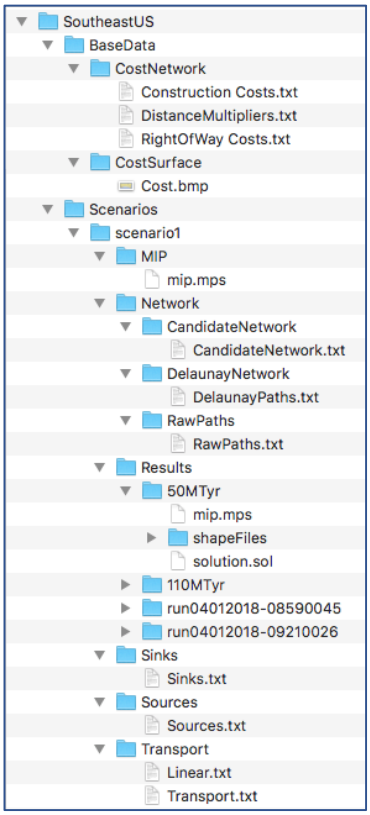
The Solver class contains the algorithms for building networks and manipulating graph representations. New network generation algorithms would belong here.

MPSWriter

The MPSWriter class formulates the optimization problems. Custom models would be implemented here.

Data Format and Organization

SimCCS functionality is entirely dependent on being able to access CCS instance data. At a minimum, this data consists of source, storage, and cost surface data. The organization of data files expected by *SimCCS* is shown below.



Naming conventions must be followed expect for dataset names (e.g. SoutheastUS), scenario names (e.g. scenario1), and result names (e.g. 50MTyr). All other directories and files must be in the same location and named the same thing as shown above.