StorageBOSSE Manual

# Overview

StorageBOSSE (Storage Balance of System Systems Engineering) is a is a semi process-based, ground-up model that first estimates the engineering and physical calculations of a utility-scale storage plant (> 1 MWDC), and then calculates cost estimates based on these engineering results. It adds battery costs to BOS, finding the full CAPEX of the storage plant. StorageBOSSE is based on the LandBOSSE code architecture.

Shown below is a very high-level representation of the software architecture:

Manager

**Input Data**

**Output** **Data**

Site Prep

Collection

Foundation

Container Erection

Grid Connection

Substation

Management & Development

Layout Optimizer

Manager calls each module. Storage layout is determined in Layout Optimizer, and costs are calculated in each other module.

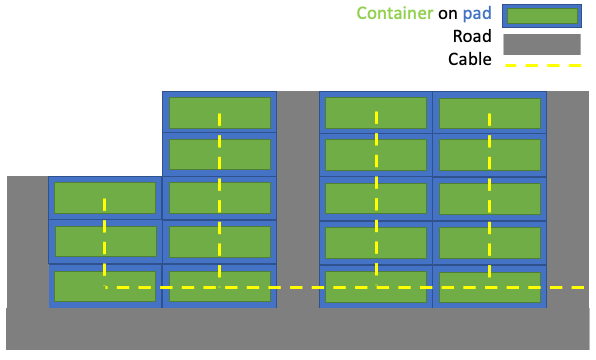
# Assumptions

* Considers battery storage (other technologies can be added in future work)
* Batteries are sold in containers, which also contain an inverter and transformer (this is a common setup). User specifies container cost and dimensions.
* AC collection system
* Road and (underground) cable extends from substation to container grid.
* Substation and Grid connection costs are based on curve fits for $ vs kV (same as landBOSSE)
* Foundation/Site Prep/Container Erection costs are process based
* Collection cost is based on landBOSSE runs. Uses lumped $/m.
* Management & Development costs are lumped. Used PV + Storage Cost Benchmark study by Ran Fu1, et al to determine exponential curve-fit of $ vs MWh. Lumped costs consist of:
  + EPC overhead
  + Developer overhead
  + EPC/Developer net profit
  + Contingency
  + Sales tax

# Layout Optimizer

Modes:

* In Linear mode, containers are stacked in one row (minimum cost layout)
* In Custom mode, the user defines the specific row layout
* In Aspect ratio optimized mode, the code finds the layout with minimum area aspect ratio layout.
* Roads wrap around double sets of rows, so that each pad is accessible by road.
* Cables start one half pad length away from the first pad in the array, and connect to the center of each pad.



# Inputs & Outputs

Inputs are stored in excel files. A project list file, with parameters for each run, and a project data file within a folder named *project\_data*

The project data file has a variety of sheets with input information, which can be modified by the user if desired. The stock file is *project\_data\_defaults.xlsx*.

The project list file (stock titled *test\_project\_list.xlsx*) contains inputs that are more likely to be modified. **Inputs** of note are:

* **Project data file**: name of project data file
* **Layou**t: defines layout type
  + - Linear - containers are stacked widthwise in one row
    - Aspect ratio optimized - containers are fit into a rectangular grid that minimizes aspect ratio
    - Custom - user defines container grid by three inputs:  
       **Number of rows  
       Containers per row  
       Containers left over** (forming a partial row, as seen in graphic)
* **DC to AC ratio**: (DC system rating) / (AC system rating)
* **site\_prep\_area\_m2:** defined project area for site preparation. If unspecified, site prep area is  
  calculated based on number of containers and road length

Each input is read into *input\_dict,* whose keys can be manually overrun by the user if desired. *Input\_dict* keys can be found in *create\_master\_input\_dict.py*

Outputs are stored in *output\_dict.* There are a host of outputs, which are read into dictionaries *BOS\_results* and *detailed\_results* in the *main.py* script. Some of note are:

total\_cost

num\_containers

total\_container\_cost

total\_bos\_cost

substation\_cost

total\_transdist\_cost

total\_foundation\_cost

total\_collection\_cost

total\_management\_cost

total\_roadcost

total\_erection\_cost

# Running StorageBOSSE

StorageBOSSE is currently located in the *StorageBOSSE* branch of *hybrids\_shared\_infrastructure,* located at this git repository: <https://github.com/parangat94/hybrids_shared_infrastructure>.

StorageBOSSE will be made pip installable. In the meantime, it can be run from *./StorageBOSSE/main.py*. The three lines required to run StorageBOSSE are:

Input\_dict = dict()

input\_dict['project\_list'] = 'project\_list\_test'   
BOS\_results, detailed\_results = run\_storagebosse(input\_dict)

The second specifies the name of the project list excel file, which must be at the same level as *main.py*

*Main.py* has an example of varying system size (MW DC and MWh), and outputting data to an excel file.

*1https://www.nrel.gov/docs/fy19osti/71714.pdf*