***OptSandbox description***

1. Source Data and Base Condition tables are read in. [TblLoader]
2. Optimization Instance options are parsed (either from table or gui) [OptionLoader]
   1. **Agencies**
   2. **Sectors**
   3. **Geographic scale**
   4. **Geographic area**
   5. **Base year**
   6. **Base condition**
   7. **Wastewater data**
   8. **Cost profile**
3. Check to make sure the metadata options are valid (i.e. present in the SourceData) [OptionLoader]
4. Geographic and non-geographic entities are extracted from SourceData, so that they can be used as a logical mask [IncludeSpec]
   1. Determine the complete list of geographic entities (using LRsegs, CountyName, StateAbbreviation, StateBasin) to be included in this Optimization Instance
      1. Generate Boolean mask for the dataframe based on the option specifications
      2. A logical OR amongst the geographic options is computed.
   2. Determine the complete list of non-geographic entities (using agencies) to be included in this Optimization Instance
      1. Generate Boolean mask for the dataframe based on the option specifications
5. A matrix is generated with rows=(LRseg, agency, source) and cols=BMPs [PossMatrix]
   1. Geo and non-geo entities are used to query BaseCondition data and get the LoadSources (**along with their maxes**) for each LRseg-agency coordinate [SegmentAgencyTypeFilter]
      1. Load sources (*for natural, developed, agriculture, and septic*) within the specified LRseg-agency coordinates are found by filtering with the Geo Boolean mask and then the agency Boolean mask
      2. Load sources (*for animals and manure*) within the specified LRseg-agency coordinates are found by filtering with the Geo Boolean mask (**filtering by agency is not needed for animals and manure**)
   2. Three sparse matrices are created [(LRseg, Agency, Source) x BMP]

* *For the Land table, the specs are rows=seg-agency-sources X columns=BMPs*
* *For the Animal table, the specs are rows=FIPS-animal-sources X columns=BMPs.*
* *For the Manure table, the specs are rows=FIPSto-FIPSfrom-animal-sources X columns=BMPs.*
  + 1. All the possible FIPSFrom and FIPSTo combinations are generated.
  1. A series of all the LoadSource in this Optimization Instance is created.
  2. A dictionary is generated mapping Loadsources -> Eligible BMPs
  3. NonNaN markers are inserted into the Possibilities Matrix for eligible (Geo, Agency, Source, BMP) coordinates

1. Populate the Possibilities Matrix at eligible (Geo, Agency, Source, BMP) coordinates with numbers less than their Hard Upper Bounds
2. Reformat Possibilities Matrix to Multi-index vector
3. Write Multi-index vector to tab-delimited text file

