# Contents of README file

# I. COPYRIGHT AND THIRD PARTY SCRIPTS
# II. MANUAL
# II.0 How to update folder definitions
# II.1 How to update configuration file
#
# I. COPYRIGHT AND THIRD PARTY SCRIPTS
#
# Authors of the project are as follows:
# Oytun Babacan (Graduate Student Researcher) obabacan@ucsd.edu
# William Torre (Principal Investigator) wtorre@ucsd.edu
# Jan Kleissl (Principal Investigator) jkleissl@ucsd.edu
#
# Contributors to the code repository with their dispatch algorithm
# Ryan Hanna (Graduate Student Researcher) rehanna@ucsd.edu
# Anders Nottrott (Graduate Student Researcher) anottrot@ucsd.edu
#
# Please see esss_master for Copyright disclaimer
#
# Please see Toolbox for Third Party scripts.
# Last known working URLs for these scripts:
# (1) datenum_round_off function
# http://www.mathworks.com/matlabcentral/fileexchange/26374-round-off-
dates-and-times/content/datenum_round_off.m
# (2) units function
# http://www.mathworks.com/matlabcentral/fileexchange/4093-units-
m/content/units.m
#
# Please see Toolbox/battery_ryan_etal_2014 for Battery Dispatch
# algorithm:
# Related publications for this algorithm can be found through the
# following DOI:
# > 10.1016/j.renene.2012.12.036
# > 10.1016/j.solener.2014.02.020
#
# II. MANUAL
#
# Master execution file of this code repository is "esss_master"
#
# Follow the following order to run simulations:
# 0) Install the latest version of MATLAB
# (http://www.mathworks.com/products/matlab/)and OpenDSS
# (https://sourceforge.net/projects/electricdss/) software. Older versions
# of these software are usually compatible with this code repository.
# 1) Update esss_getConfig.m with correct local folder definition.
# 2) Update simOverview.xlsx with desired parameters.
# 3) In MATLAB, execute def_addpath.m to add code repository to your
# MATLAB workspace.
# 4) In MATLAB, execute esss_master through MATLAB command line, e.g.
# "esss_master(1,'list_1');". The algorithm will iterate the solution sets
# until at least one convergence criteria is satisfied. See "maxGen",
# "xlen", "xtol1" and "xtol2" to review the termination criteria. The
output will be stored under "simulation" folder. This folder will be automatically created in the same directory that the algorithm files are located. All results are recorded through esss_master function. Below is a description for each result files that will be saved.

> All results regarding the genetic algorithm optimization result files can be reviewed through the data set "overview_GA". This data set stores objective function and fitness values for each solution in each iteration. Rows indicates different solutions. Columns will give the following values from left to right in order: 1_fitness value 2_number of batery systems 3_total battery capacity [kWh] 4_total battery rating [kW] 5_cost penalty, (for single date simulation) 6_loss reduction term 7_voltage deviation reduction 8_operational lifetime (for multiple date simulation) 6_7_8 will be given for each date in order. If the objective function lacks any of the terms, they will be marked as NaN.

> Sizing and siting results can be reviewed through the data set "overview_Sizing" and "overview_Siting" respectively. These data sets store the sizing and sizing decisions for the best solution among each iteration. Each row will represent each iteration. Column size is determined by the number of permissible nodes for siting defined by the user. (e.g. 20 iterations for 7 permissible nodes will result in 20x7 matrix).

> The details for the best solution can be reviewed through "overview_bestInd". This variable is a struct and the struct fieldnames are self explanatory.

# II.0 How to update folder definitions
# > Update "localFolder" variable in esss_getConfig.m function, Line 24
# > If new data sets (circuit, demand data, solar data) are to be imported, update getPathDefinitions.m script accordingly.

# II.1 How to update configuration file
# Please see below for the complete list of parameters that can be updated in simOverview.xlsx configuration file.

# Colored columns are flexible parameters that can be varied by simply changing values in the configuration file.
# White-colored columns are parameters that require additional steps in order to be varied. They should only be changed e.g. when loading new data sets, new circuit topologies, or for development purposes.

# Some examples are already given in the configuration file.
# 909: Default and well-tested values.
# 919 - 949: Example variations in parameters in each colored columns.

# Explanation of each parameter:
# The in-line comments given in esss_getConfig.m, esss_master.m and related functions and scripts should give enough guidance how to define each parameter. For convenience, a short description of each parameter is also given below.

# Colored parameters:
# simDate: Date of the simulation. Multiple date inputs are allowed. set wM (weather multiplier) to 1 for single dates. Otherwise, assign a weight
to each day, e.g. for equal importance among two dates, set wM to "0.5, 0.5"
# J: Objective function definition. Full set of objective function consists of 4 parameters, thus "1m,2m,3m,4m" will result in full objective function execution. If certain objective parameters are desired to be omitted, it can be done through this parameter definition, e.g. "1m,2m,4m" will result in omitting battery degradation parameter. The available parameters are as follows: (1m) Energy Loss Reduction (2m) Voltage Deviation Reduction (3m) Battery Operational Lifetime (4m) Battery Cost Penalty
# siting mode: Set of nodes that will be considered in the optimization. Please see circuit_findAllowedNodes.m for details. Options: Full, Restricted, Minimal. Default: Minimal.
# b: Sizing penalty parameter in the penalty cost function (4m) Default: 0.090
# c: Siting penalty parameter in the penalty cost function (4m) Default: 0.005
# set: The set of nodes where a PV system is installed. If no existing PV penetration scenario is available, a new set will be created. The user can trigger creation of new sets by simply changing the name of the set, e.g. change "s1" to "s2". The PV penetration sets are then created by 10% increments up to 100%. To allow more refined penetration scenario generation, see
# PVp: level of PV penetration in the distribution system. 10%-100% with 10% increments. Other values can be given if opendss_randomizePVlocations.m function is updated accordingly.
# rate2: Mutation rate of "bit string mutation" of the genetic algorithm mutation operator. Default: 0.010
# White-colored parameters
# id: Row ID number of the simulation. It can be different from simulation ID if desired.
# circuit: Distribution system network to be simulated. Other circuits than IEEE8500 will need updating of many scripts and functions in the repository. Please skim through the files and see the "if" clauses that are marked with IEEE8500 tag. These clauses should be written for the new distribution system. Appropriate warnings and errors will be also thrown when the user tries to simulate another distribution system. These indicators can help the user to import a new circuit.
# load data: Demand data for load points in the circuit. openEI indicates demand data imported from open energy information: openei.org
# solar data: Solar generation data for PV systems in the circuit. myPVdata indicates San Diego solar generation data.
# simID: Simulation ID number of the simulation. This is the number that should be called when executing esss_master(simID,simSheet) command to start simulations.
# wM: Weather Multiplier. Please see simDate to see how to define wM.
# a: Minimum permissible battery size that is to be installed in the circuit. [kWh] A parameter of the cost penalty function (4m). This value can be as small as 1kWh.
# s: Unit conversion and scaling factor of the cost penalty function (4m). This should be changed unless the qualitative cost penalty model is being remodelled.
# Vdev: Voltage deviation calculation type. REF indicates that voltage deviation is calculated against 1PU at each node at all times. Other options are available in esss_computeVdev.m. However, they are given for experimental purposes and should not be used in default mode.
# demand: The demand profile variation among different load points. Uniform indicates that every load point has the same demand curve shape (scaled to individual peak demand magnitudes)
# bess node: The type of nodes that will be considered in the allocation simulation. Primary node indicates that only MV lines on the primary side of the service transformers will be considered as permissible.
# ghimap: Spatial variation in solar generation data. Uniform indicates that all PV systems have the same solar generation curve.
# elitism: The percentage of the GA population that will be considered as elites. Default: 2%
# selection: The type of GA selection operator.
# k: selection pressure of the GA selection operator. [0-1] Default: 0.80
# crossover: The type of GA recombination operator. PUC indicates parameterized uniform crossover.
# Po: recombination pressure of the GA recombination operator. [0-1] Default: 0.50
# mutation1: The first phase of the GA mutation operator. Random generation indicates, new individuals will be created during this first phase. If two-phase mutation is not desired, this behavior can be cancelled through mtrigger.
# rate1: The mutation rate of the first phase GA mutation operator. In the case of random generation, the number indicates the number of individuals that will be randomly generated at each iteration. It should be less than the number of individuals in the population. Default: 10
# mutation2: The second phase of the GA mutation operator. Bit string mutation introduces bit inversions in each solution set subject to a mutation probability.
# rate2: The mutation rate of the second phase GA mutation operator. Higher values will trigger more bit-string inversions in each iteration. Default: 0.010
# mtrigger: The transition from the first phase to the second phase of the GA mutation operator. Set to 1 if this behavior is not desired. Default: 5
# maxGen: (Termination condition) Maximum allowed iterations in the GA optimization. Default: 100
# xlen: (Termination condition) Maximum allowed iterations with no fitness change in the best individual of the population. Default: 10
# xtol1: (Termination condition) The amount of fitness change that is desired in the best individual of the population between each iteration, so that xlen condition is reset. Otherwise, no fitness change will be assumed and when the count reaches xlen, the optimization will be terminated. Default: 0.001
# xtol2: (Termination condition) Change in the number of battery systems installed in the best individual of the population between each iteration. If, no change is observed for 2*xlen, the optimization will be terminated. Default: 1
# numY1: Minimum number of battery systems that is to be installed to an individual solution in initial population generation. Default: 1
# numY2: Maximum number of battery systems that is to be installed to an individual solution in initial population generation. Default: 7
# capY1: Minimum cumulative battery system capacity that is to be installed to an individual solution in initial population generation. Default: 5000 [kWh]
# capY2: Maximum cumulative battery system capacity that is to be installed to an individual solution in initial population generation. Default: 20000 [kWh]
# minCap: Minimum allowed battery capacity in any of the nodes. Defined as the percentage of the peak demand of the circuit. When this minimum is observed at any node, that system will be discarded from the solution. Default: 0.001
# timeStep: Time step of the demand and solar data. Also defines the timestep of the steady state power flow simulations. Default: 900s
# critNode: The scaling parameter to determine total number of critical nodes to consider in the optimization. By default, the number of critical nodes equals to the number of PV systems. Default: 1
# repPerc: The maximum percentage of population that is to be replaced with new individuals when the power-flow simulation convergence is not achieved due to extreme battery system capacities at that node. The reason that new individuals have to be initiated is that the dispatch algorithm does not have power-flow constraints. Default: 0.20 [%]
# simPVtype: The PV penetration scenario generation approach. Match demand indicates that for each PV system, the capacity of PV system will match the peak load demand at that node. Default: customer-matching
# simBESSdist: If set to "equal", total battery capacity is distributed equally among initialized nodes. Otherwise, they will vary according to the rules set in esss_initializePopulation.m function. Default: varying
# simFlag: If set to "none", full binary set will be used to determine battery capacity (resolution of 1kWh). By setting to "exhaustive", the resolution of battery capacity can be determined. Please see esss_master.m function for details. Default: exhaustive