Study of Co-optimization Stochastic SuperOPF Application in the CAISO System

Design Manual

Prepared for

Prepared by
Bigwood Systems, Inc.
Ithaca, NY

March 30, 2015

Design and Implementation of the SuperOPF Program

Bigwood Systems, Inc.
35 Thornwood Dr., Suite 400, Ithaca, NY 14850

1. Background

The optimal power flow (OPF) is relevant in power system operations, scheduling, and planning. The main objective of the OPF problem is to determine the optimal steady-state operation of an electric power system while satisfying technical and economic constraints. With the structural deregulation of electric power systems and prevalence of power markets, OPF is becoming a basic tool for planning and operations of the power network.

Existing solvers and programs for OPF computation are mostly focused on achieving an optimal solution to the study objective for a single scenario, namely, the base case system, without considering any uncertainties that might arise during the planning horizon. In fact, with more and more penetrations of renewable energy, whose output is stochastic in nature, in the power networks and requests for regulatory compliance (such as compliance to FERC regulations), these uncertainties cannot be neglected any longer. It is crucial for the OPF solutions to not only be economic for the base-case system, but also maintain economic and secure for the whole planning horizon even if one or more uncertain scenarios (with the largest probabilities) occur. In the meantime, it is also crucial for the OPF solutions being not only economic and but also secure, such that the power network can survive the occurrence of contingencies where single or multiple devices in the network go offline suddenly.

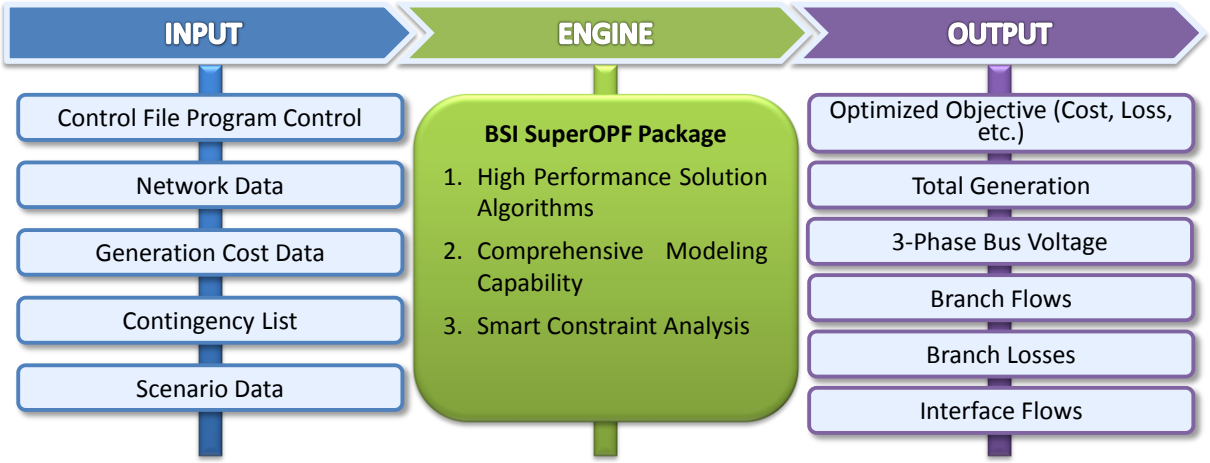

Figure 1. A schematic of the SuperOPF procedure

This motivates the development of the multi-scenario security-constrained SuperOPF solver and program, to address the need of computing OPF solutions involving contingency constraints and uncertain scenarios for power networks nowadays that are operated in a more

and more dynamic environment. This document details the design and implementation of the SuperOPF program. A schematic diagram of the SuperOPF procedure is provided in Figure 1.

2. Multi-Scenario Optimal Power Flow Analysis

Under the stochastic co-optimization framework of SuperOPF, this stage will develop the single-period, stochastic optimal power flow (OPF) solver and program to deal with uncertainties of the operating conditions which will be expected at the decision-making moment or during the planning horizon. These uncertainties encountered during the planning horizon can be categorized into two types. The first type of uncertainties relate to discrete events such as failure of devices (lines, generators, shunts, etc.), that is, contingencies. The so-call (N-1) criteria, that is, there is only one device of the power network fails for a single contingency, will be considered in the stochastic problem formulation. The other type of uncertainties stems from limited knowledge about future model parameters, for example, uncertainty in the forecasts of load, climate, wind or river flow, which are stochastic in nature. Such uncertainties become increasingly non-neglectable because of the increasing penetrations of renewable energy in the power networks.

In the spirit of SuperOPF, these types of uncertainties will be described as a set of probability distributions for the uncertain parameters and events and will be materialized as multiple scenarios (sets of system states) with associated probabilities. Combination of contingency and stochastic scenarios will result in a tree-like structure of system states with associated probabilities, where the contingent events provide a set of base scenarios while the stochastic consideration adds a set of stochastic scenarios to each base scenario. The objective of the OPF computation is thus a probability-weighted sum of the objectives of the materialized scenarios. The resulting OPF solver and program will be able to co-optimize these multiple materialized scenarios to realize the desired stochastic OPF computation, such that an optimal operating plan can be reached at which the total expected operating objective is optimized (such as the system production cost is minimized, or the total expected net benefit is maximized, or the system power losses are minimized), while all physical and operational limits imposed on the power system are satisfied (such as, no voltage violations and no thermal violations).

3. The SuperOPF Solution Framework

The central task for the SuperOPF solver is to solve the optimal power flow problem which will result the best expected objective (such as the minimum system total power losses or generation costs) spanning the planning horizon. Therefore, the co-optimization that needs to be solved in order to get the desired optimal power flow solution can be modeled as the optimization problem (1).

It can be seen from the problem model (1) that the optimization problem that needs to be solved is a very complicated nonlinear optimization problem. The problem size (the number of optimization variables and the number of equality and inequality constraints) will increase linearly as the number of internal scenarios increases.

$$
\begin{aligned}
min \quad & f(x) = f_0(x_0) + \sum_{k=1}^{K} p_k[f_k(x_k) + c_k(x_k - x_0)] \\
s.t. \quad & h_0(x) = 0 \\
& g_0(x) \leq 0 \\
& \cdots\cdots \\
& h_K(x) = 0 \\
& g_K(x) \leq 0
\end{aligned}
\tag{1}
$$

3.1 The Overall Framework

An overall structure of the SuperOPF solution framework is presented in Figure 2. The solver takes several types of input data, namely, the base case power flow data, the contingency list, and the renewable energy forecasts. Based on the input data, internal scenarios will be formatted in a tree-like structure. For each of these internal scenarios, an internal nonlinear optimization (NLP) model is constructed, along with the master optimization model associated with the co-optimization problem spanning the whole set of scenarios. All these internal optimization models are then fed to the SuperOPF co-optimization solver for computing the optimal power flow solution that achieves the best expected objective value.

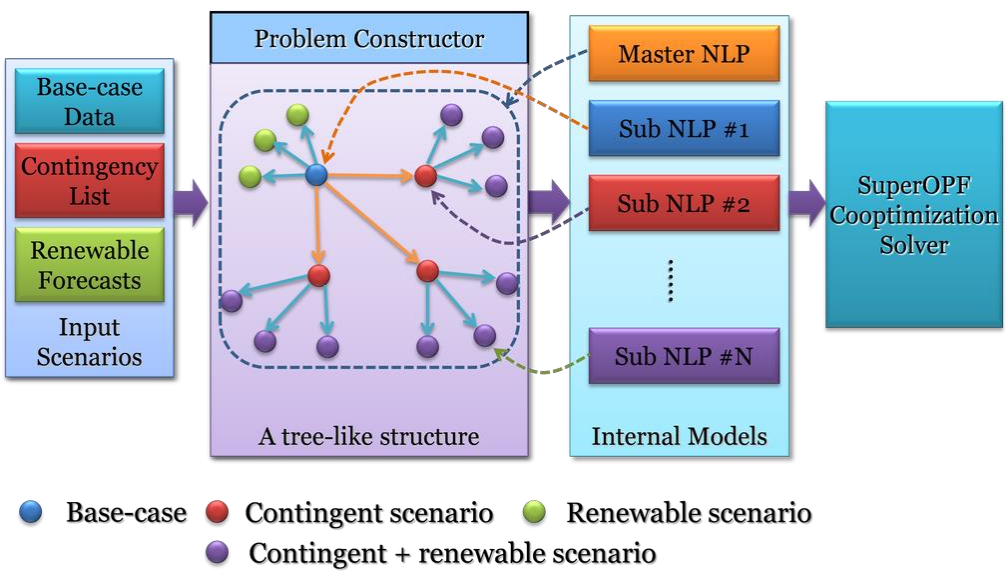

Figure 2. Illustration of the SuperOPF procedure

Internal scenarios are categorized into four types, namely, type-1 through type-4 scenarios. Format of these four types of scenarios are summarized as follows:

- Type-1 base case problem where there is no any contingency in the power network and not consider renewable energy outputs.

Type-1 scenario:

$$\begin{aligned} &min && f(x) \\ &s.t. && P_i(x) + P_{Di} - P_{Gi} = 0 \qquad 1 \le i \le n_B \\ & && Q_i(x) + Q_{Di} - Q_{Gi} = 0 \\ & && S_k = \sqrt{P_{ij}^2(x) + Q_{ij}^2(x)} \le S_k^{max} \qquad (i,j) \in L \\ & && x^{min} \le x \le x^{max} \end{aligned}$$

where, n_B is the number of buses and L stands for the set of branches.

- Type-2 scenario problems, where there is a single contingency involved in the power network.

Type-2 scenario:

$$\begin{aligned} &min && f(x) \\ &s.t. && P_i(x) + P_{Di} - P_{Gi} = 0 \qquad 1 \le i \le n_B \\ & && Q_i(x) + Q_{Di} - Q_{Gi} = 0 \\ & && S_k = \sqrt{P_{ij}^2(x) + Q_{ij}^2(x)} \le S_k^{max} \qquad (i,j) \in \hat{L} \\ & && x^{min} \le x \le x^{max} \end{aligned}$$

where, $\hat{L}$ is L excludes contingent branches.

- Type-3 scenario problems, where there includes a single set of forecasts of the renewable energy generations.

Type-3 scenario:

$$\begin{aligned} &min && f(x) \\ &s.t. && P_i(x) + \hat{P}_{Di} - P_{Gi} = 0 \qquad 1 \le i \le n_B \\ & && Q_i(x) + \hat{Q}_{Di} - Q_{Gi} = 0 \\ & && S_k = \sqrt{P_{ij}^2(x) + Q_{ij}^2(x)} \le S_k^{max} \qquad (i,j) \in L \\ & && x^{min} \le x \le x^{max} \end{aligned}$$

where, $\hat{P}_D$ and $\hat{Q}_D$ are equivalent loads with renewable energies.

- Type-4 scenario problems, where the combinatorial occurrence of a contingency and a renewable forecast is considered.

Type-4 scenario:

$$\begin{aligned} &min && f(x) \\ &s.t. && P_i(x) + \hat{P}_{Di} - P_{Gi} = 0 \qquad 1 \le i \le n_B \\ & && Q_i(x) + \hat{Q}_{Di} - Q_{Gi} = 0 \\ & && S_k = \sqrt{P_{ij}^2(x) + Q_{ij}^2(x)} \le S_k^{max} \qquad (i,j) \in \hat{L} \\ & && x^{min} \le x \le x^{max} \end{aligned}$$

Assuming there are M contingencies in the input contingency list and N renewable forecasts, then the total number of internal analysis scenarios will be (M+1)*(N+1). The numbering of the internal scenario for a session with 3 contingencies and 2 renewable forecasts is illustrated in Table 1.

Table 1. Numbering of the internal scenarios

Scenario Index	Base case	Contingency 1	Contingency 2	Contingency 2

Base case	1 (type 1)	2 (type 2)	3 (type 2)	4 (type 2)
Forecast 1	5 (type 3)	6 (type 4)	7 (type 4)	8 (type 4)
Forecast 2	9 (type 3)	10 (type 4)	11 (type 4)	12 (type 4)

4. Implementation

The SuperOPF program has a modularized structure and is designed to be ready for future extensions. It is flexible and convenient for further development, in order to support more data formats and to enclose other effective linear and nonlinear solvers into the current implementation. A block diagram for the structure of SuperOPF program is shown in Figure 3. The program is composed of two major parts, that is, the solver modules and the user interface programs.

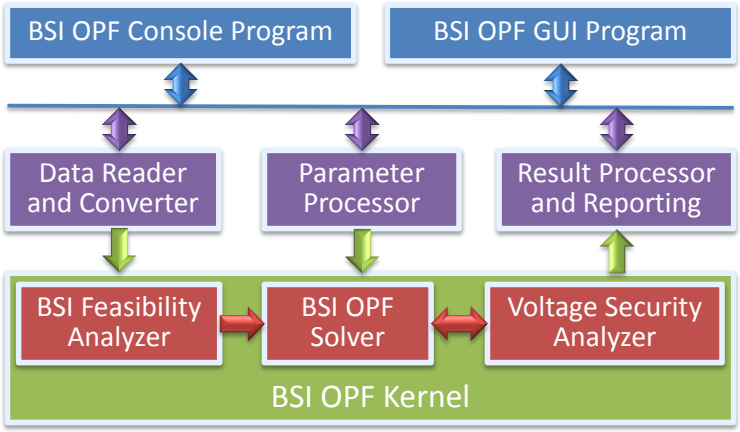

Figure 3. SuperOPF program structure

4.1. The SuperOPF Solver Modules

The kernel part is designed to take over all the computationally intensive and architecture independent tasks. These tasks mainly include data file reading, parameter settings, result presenting and output file writing, and SuperOPF computations. To this end, the solver modules of the SuperOPF program can be divided into two categories, that is, the data input/output (I/O) and representation modules and the kernel computing modules.

The data representation modules handle the tasks of data file reading and converting required by the SuperOPF program, parameter settings for computation, and result presenting. Therefore, this category consists of components for reading power flow files and other data files. The most important data file to be processed is the power flow file, where the structure of the power network under study, parameters of the involved network components, and the initial state of the power network are specified. There are many data formats used by different vendors and utilities in the power industry, among which PSS/E, PSLF, PSF, and CIM are the most popular

ones. In the SuperOPF program, different power flow data I/O modules are implemented, each of which is dedicated to one data format. All these data I/O modules are derived from the power flow I/O base module. External power flow files in different formats are read and processed by the I/O engine and then converted to an internal, unified power network representation. In such a way, efforts for future support of other data formats can be minimized. The data file I/O engine implements the procedure of reading other data files following the same philosophy of flexibility and extensibility. The other data files need to be processed in the SuperOPF program include the generation cost model file for minimizing the system generation cost and the contingency related files for scenarios of OPF with security constraints. Another important data file for SuperOPF program is the scenario specification file, which specifies the list of scenarios to be analyzed by the program. Besides the data reader and converter module, the data representation category also includes a module for handling parameter settings which interprets the user configured parameters and converts them to internal representations which is understandable by the compute engine. The last module in this category is the module for result data representation, which is designed to interpret and archive the output files produced by the compute engine.

The actual multi-scenario optimal power flow computation and voltage security analysis are handled by the kernel computing modules. All the data collected and processed by the I/O engine is fed to the compute data module of the compute engine. This data module defines all data structures for hosting data to be used during the actual optimization process. Necessary transformations of the data fed from the I/O engine are carried out in this module. Computation parameters are processed by the parameter setting module. The data and parameters are then fed to the OPF solver module and the actual OPF computation is executed. The first module of the SuperOPF core engine is a feasibility analyzer, which is designed to compute a feasible point if the study model is feasible or to compute a restored feasible point with an optimal constraint relaxation if the study model is infeasible. The second module, SuperOPF solver, and the third module, voltage security analyzer, then use computed feasible point as the initial condition to carry out a combined optimal power flow computation and voltage security analysis, which will produce in a secure optimal power flow solution.

The SuperOPF program is designed to support different effective solvers, such as the interior point method (IPM) solver and the sequential quadratic programming (SQP) solver. To this end, the central part of the computation engine is the OPF solver base module (NLP), where the general nonlinear program modeling and solver calling are implemented. All solver modules for realizing specific optimization methods or optimizers are derived from this base module. Furthermore, the SuperOPF kernel part is implemented in C++ language using only standard, architecture independent libraries. Therefore, the kernel part can easily be ported to other architectures and operation systems with minimum efforts. The SuperOPF interfaces are developed in C++ with Microsoft MFC libraries and C#, which provide natively-supported and convenient resources for developing a user-friendly and feature-rich GUI in Microsoft Windows environments for the SuperOPF program.

4.2. SuperOPF User Interface Programs

The user interface part also consists of two components. The first component is a graphical user interface (GUI) program. This GUI program provides the user a convenient and feature-enriched interface to interact with the underlying OPF computation.

There are three major functional parts in this GUI program, that is, the data selection and display part, the parameter and model setting part, and the resulting reporting part. The first part, that is, the data selection and display part handles the task of selecting required data file paths for the OPF computation. The usual data files required for an OPF computation include the power flow file and generation cost file (when the OPF objective is to minimize the system generation cost). The selected data files will also be displayed by the GUI program in an organized way and can be easily reviewed or modified by the user. The second part of the GUI program is for parameter and model settings. Responsive interfaces are designed for the user to set up a desired OPF computation scenario by specifying the problem model (the optimization objective, the cost model, etc.) and editing computation parameters (the optimization strategy, the optimizer to use, the detailed optimization parameters, etc.). Therefore, the user has a full control over the computation to be carried out. The third part handles the result reporting. Taking advantage of existing reporting engines, namely Microsoft Reports, feature-enriched and meaningful representations of the OPF computation results can be automatically produced and reported to the user.

Besides the GUI program, a SuperOPF console program is also considered in the implementation. This console program eliminates all graphical interactions and can only be run in a command line environment. However, the user still has full control over the computation scenarios through specifying the parameters to the SuperOPF console command. This results in a stand-alone, lightweight SuperOPF program suitable for low-end hardware environments. Moreover, such command-line based execution of the SuperOPF program provides the user a convenient way to effectively cooperate with other computation and management programs. For example, the user can include the call of the SuperOPF console program in a script to automate his/her analysis tasks of sequential execution of multiple programs or the task of analyzing a batch of scenarios.

4.3. SuperOPF Class Diagram

A diagram for relationship between the major classes constituting the SuperOPF solver kernel part is shown in Figure 4.

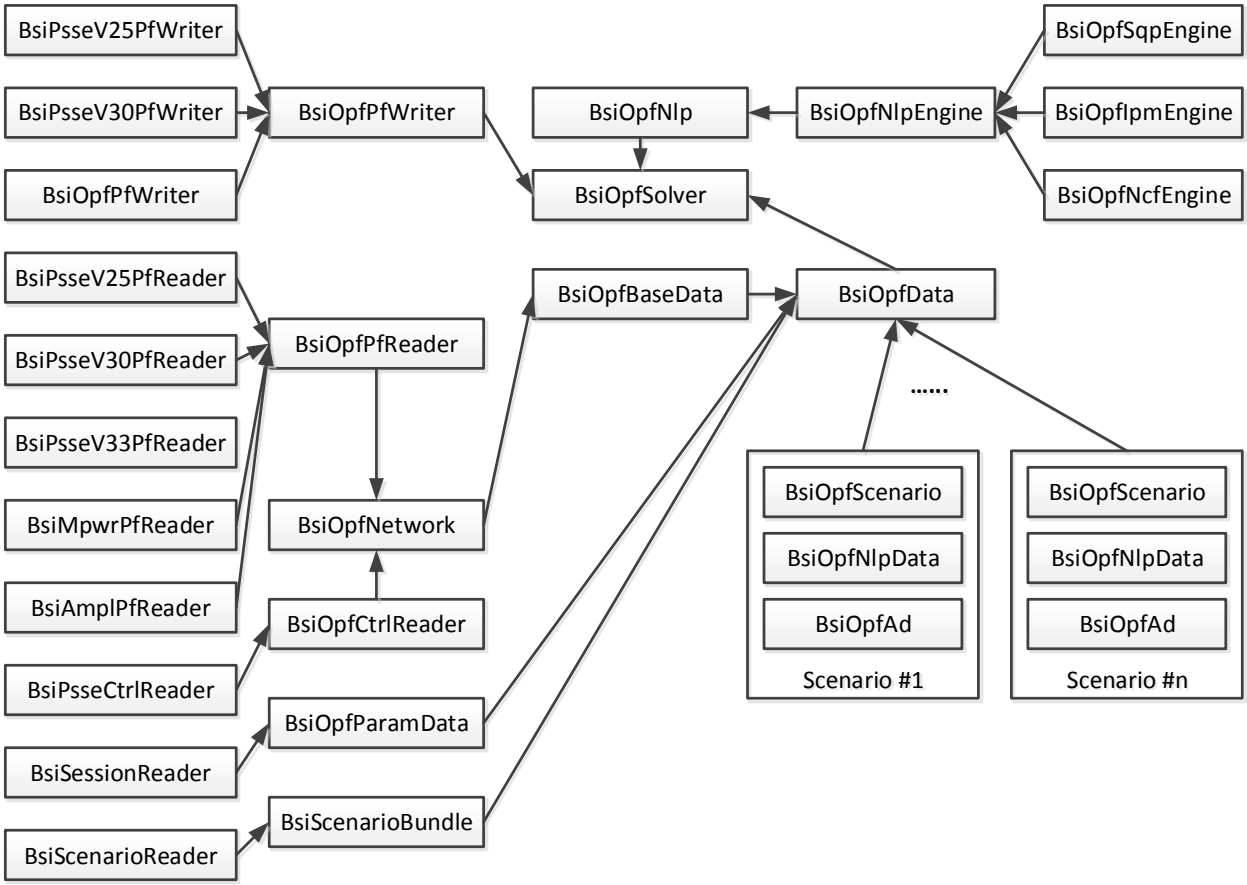

Figure 4. A class diagram for the multi-scenario SuperOPF solver

Functionality of these classes is described as follows:

- *BsiOpfPfReader*: this class is the base class for reading power flow data from different formats of files.
- *BsiPsseV25PfReader*: this class implements the functionality to read and interpret power flow data formatted in PSS/E version 25.0 power flow files.
- *BsiPsseV30PfReader*: this class implements the functionality to read and interpret power flow data formatted in PSS/E version 30.0 power flow files.
- *BsiPsseV33PfReader*: this class implements the functionality to read and interpret power flow data formatted in PSS/E version 33.0 power flow files.
- *BsiMpwrPfReader*: this class implements the functionality to read and interpret power flow data formatted in Matpower (both version 1 and 2) power flow files.
- *BsiAmplPfReader*: this class implements the functionality to read and interpret power flow data formatted in AMPL modeling data files.
- *BsiOpfNetwork*: this class implements the internal and unified representation of the power networks model.

- *BsiSessionReader*: this class implements the functionality to read and interpret session files for the program. A session file stores the set of necessary parameters to support a SuperOPF study.
- *BsiOpfParamData*: this class stores all the parameters the will be involved in the SuperOPF computations.
- *BsiOpfCtrlReader*: this class implements the base class to read and interpret the OPF control data.
- *BsiPsseCtrlReader*: this class implements the functionality to read and interpret the OPF control data in the PSSE OPF raw data format.
- *BsiScenarioReader*: this class implements the functionality to read and interpret the scenario specification file.
- *BsiScenarioBundle*: this class stores information of the bundle of scenarios read from the scenario specification file.
- *BsiOpfBaseData*: this class stores some basic/common data structures that will be used in the SuperOPF solver, such as the network data.
- *BsiOpfScenario*: this class implements the internal scenario representation that is actually involved in the SuperOPF computation. Each scenario represents the base case, a single contingency, a single renewable forecast, or the combination of a single contingency and a single renewable forecast.
- *BsiOpfNlpData*: this class stores the data structures (such as the variables, first- and second-order derivatives, sparsity structure of the problem, etc.) that will be used in constructing the nonlinear optimization problem associated with an internal scenario.
- *BsiOpfAd*: this class implements the functionality of computing the objective and constraint function values, their sparsity structure (positions of nonzero elements in their first- and second-order derivatives), and their first- and second-order derivatives for a given set of variable values.
- *BsiOpfData*: this class stores all the data structures that are involved in the SuperOPF computation.
- *BsiOpfSqpEngine*: this class implements the improved sequential quadratic programming (SQP) algorithm for solving a nonlinear optimization problem (such as the sub-problem associated with each internal scenario and the master co-optimization problem) involved in the SuperOPF computation.
- *BsiOpfIpmEngine*: this class implements the improved interior point method (IPM) algorithm for solving a nonlinear optimization problem (such as the sub-problem associated with each internal scenario and the master co-optimization problem) involved in the SuperOPF computation.
- *BsiOpfNcfEngine*: this class implements the improved nonlinear complementarity function (NCF) method for solving a nonlinear optimization problem (such as the sub-problem associated with each internal scenario and the master co-optimization problem) involved in the SuperOPF computation.

- *BsiOpfNlpEngine*: this class is the base class for the above different solution algorithms that are involved in the SuperOPF program. This class handles common tasks and data structures for the actual solution algorithms.
- *BsiOpfNlp*: this class implements a unified interface for defining different optimization problems involved in the SuperOPF computation in the unified format of nonlinear program representation.
- *BsiOpfPfWriter*: this class is the base class for writing the SuperOPF computational results to external files of different formats.
- *BsiPsseV25PfWriter*: this class implements the functionality of writing the OPF results to a power flow file formatted in the PSS/E version 25.0 specifications.
- *BsiPsseV30PfWriter*: this class implements the functionality of writing the OPF results to a power flow file formatted in the PSS/E version 30.0 specifications.
- *BsiPsseMpwrPfWriter*: this class implements the functionality of writing the OPF results to a power flow file formatted in the MATPOWER (version 1 or 2) specifications.
- *BsiOpfSolver*: this class implements the main solver for SuperOPF.