CHAPTER 4

Computational Framework for Economic Dispatch Problem

4.1 Introduction

Generally the framework is platform which provides support for the development of a specific domain of applications. This chapter presents the design and implementation of framework targeting the economic dispatch solution.

4.2 Framework

Framework is a set of cooperating classes that makeup reusable design for a specific class of software and defines overall structure of application, and it’s partitioning in classes and objects. Architecture of a set of applications acts as basic conceptual model for framework design.

Frameworks have basic intentions and purpose for reuse. There are two common reuse strategies: white box reuse & black box reuse. White box reuse emphasizes the reuse on source code thus necessitating the understanding on source code level for developers, whereas in black box reuse, reusable components are in binary form and application developer does not access to source code. This type of reuse is normally more safe and easy to implement [75].

A framework can be customized to a particular application by creating application specific subclasses of abstract classes targeting a specific group of applications, which have similar architecture design [76]. A fairly large number of useful frameworks are available for application builders. One of these frameworks is Microsoft’s Document-View Architecture. It provides windows based menu driven application framework for data handling through documents and data visualization through views. Microsoft
Foundation Classes (MFC™) provides extensive functionality for such primitive tasks. Standard Template Library (STL) is a toolkit, which provides fine implementation of collections and common algorithms.

### 4.3 Computational Framework “PED_Frame” --- Need & Scope

Economic dispatch is vital step in electrical power system operation. It is being solved by conventional optimization techniques and artificial intelligence tools based approaches. Generally the algorithms for these techniques are implemented as stand alone programs. The common software environment integrating the various algorithms is the need for analysis in research and development activities. The Framework “PED_Frame” [77] is the platform providing common environment for implementing the algorithms for economic dispatch solution. This framework has the potential and ability to reuse the existing code. The version one of the “PED_Frame” targets the implementation of traditional, GA based and hybrid methodologies for economic dispatch used in the present research work.

### 4.4 Features and Characteristics of Proposed Framework

In economic dispatch solution many common requirements are encountered in most of the algorithms. PED Frame provides basic structure to fulfill these requirements and exposes the following functionalities in such a way that all algorithms can use them easily:

1. Easy integration of new algorithm
2. Options for application of algorithms and their parameters
3. Ability to accept input data from standard format
4. Results in the form of formatted text files
5. Analysis progress visualization mechanism
6. Activity logging mechanism for view of intermediate results.
7. Handling of hybrid methodologies
8. Comparisons of results
9. Common services.
4.5 **PED_Frame Design**

The proposed framework “PED_Frame” targets power economic dispatch software application development. It provides white-box reuse mechanism for incorporating new algorithms. It provides application developer with capability of data input in the form of machines cost curves and other algorithm specific information, which is standard for all economic dispatch applications. PED_Frame also provides outputs in a standard format. All inputs and outputs are handled through a grid. The class diagram of PED_Frame is shown in Figure 4.1

![Class diagram of PED_Frame](image)

**Figure 4-1 Class diagram of PED_Frame**

Analysis status visualization is useful during economic dispatch calculations, therefore PED_Frame provides extremely efficient component for status visualization. As most of the electrical engineering application are using vector and matrix algebra. Standard Template Library (STL) is used as base for the development of the vector-matrix library. This library gives efficient algorithm implementation for vectors and matrices [78].

4.5.1 **MFC Document -View Architecture**

PED_Frame is primarily based upon doc-view architecture of Microsoft Foundation Class. It acts as container for PED_Frame and provides base services. These
services are primarily in the form of messaging mechanisms, and threading support. The base and derived classes are listed in table 4.1.

**Table 4-1 Classes in PED_Frame**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Class</th>
<th>Base Class</th>
<th>Remarks Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CPEDApp</td>
<td>CWinApp</td>
<td>This class manages overall framework.</td>
</tr>
<tr>
<td>2</td>
<td>CMainFrame</td>
<td>CFrameWnd</td>
<td>This class manages the main window.</td>
</tr>
<tr>
<td>3</td>
<td>CPEDView</td>
<td>CScrollView</td>
<td>This class provides basic services for visualization of application data. It also contains grid objects and manages their functionality</td>
</tr>
<tr>
<td>4</td>
<td>CPEDDoc</td>
<td>CDocument</td>
<td>This class works as data repository and its management. It also contains instances of matrix class for input matrix, B-coefficient matrix and output matrix. The reference of these matrices is passed to respective CPEDAnaslysisBase derived class, which performs analysis by using these matrices and updates output matrix.</td>
</tr>
<tr>
<td>5</td>
<td>CPEDAnaslysisBase</td>
<td>CWinThread</td>
<td>It acts as base class for different classes which implement algorithms.</td>
</tr>
<tr>
<td>6</td>
<td>CPEDAnalysisStatus</td>
<td>CDilog</td>
<td>It manages the current status of execution and visual components like progress bar.</td>
</tr>
<tr>
<td>7</td>
<td>CPEDGABase</td>
<td>CPEDAnaslysisBase</td>
<td>This class acts as base class for GA based applications</td>
</tr>
<tr>
<td>8</td>
<td>CPEDCABase</td>
<td>CPEDAnaslysisBase</td>
<td>This class acts as base class for CA based applications</td>
</tr>
</tbody>
</table>
4.5.2 Input/Output Using Grid

CPEDGrid class acts as interface for grid, which provides functionality like input, formatted output, print and data validation checks. It extends grid functionality to link its services with other parts of the PED_Frame. These are:

a. Data present in Grid is transferred to a matrix, which can act as data communicator.

b. Data present in matrix is formatted in grid for display and printing purposes.

c. Data editing and data validation checks.

PED_Frame creates three instances for grid class --- Machine data, B-coefficients and output.

4.5.3 Analysis Status Visualization

Analysis status visualization is implemented in class CPEDAnalysisStatus derived from CDialog class of MFCIM. This dialog window contains progress indicators showing current state of convergence, iteration count, and time elapsed.

4.5.4 Vector-Matrix Library

This library includes template based CPEDVector and CPEDMatrix classes. CPEDVector is derived from vector<class T> and it implements functionality for initialization, resizing, addition, multiplication, searching, sorting and debugging routines. CPEDMatrix class is derived from CPEDVector <CPEDVector <class T>> and it implements functionality for initialization, resizing, and matrix algebra operations such as addition, multiplications, division, transpose, inverse.

4.5.5 File Read Writes

The format for input and output has been standardized. The data files for all the test systems have been prepared according to this standard and data is read through these files. The results may be written with different options in output files.
4.6 Framework Implementation

Analysis is the process of computing ED solution using the techniques in the PED_Frame. This part of the framework is responsible for actual implementation of algorithm. This work is accomplished through following classes:

a. CPEDAnalysisBase class is derived from CWinThread so acting as independent thread of execution. It contains references to matrices for inputs such as power demand, machine cost curves co-efficient, B-coefficient, GA parameters, encoding schemes, GA operators, maximum iterations count, error tolerance and output. It also provides virtual functions for start analysis, communicating status information and formatting output.

b. CPEDGABased is derived from CPEDAnalysisBase and works as base for implementation of genetic algorithm. This class contains information for Genetic Algorithm (GA) parameters. A new class can be derived from this class to implement any GA based solution for Economic dispatch.

c. CPEDCABase is also derived from CPEDAnalysisBase and works as base for conventional algorithm implementation. It contains information like initial start and step length etc. A new class can also be derived from this class to implement any mathematical programming based solution for Economic dispatch.

4.7 Steps for Implementation of Customized Solutions Using PED_Frame

PED_Frame is useful if we implement customized solution based on generic services provided by it. We need to perform some specific task to use these services in customized solution.

4.7.1 Economic Dispatch Using Mathematical Programming based Approach

It can be implemented as follows:

a. Derive a class from “CPEDCABase” and declare algorithm specific parameters in it.

b. Write code for new algorithm in StartAnalyse function, which is in overridden form.

c. Call UpdateStatus with error value and current iteration.
d. Override “FormatResults” to transfer results from analysis structure to output matrix.

e. Create a new menu item and write its message handler in CPEDDoc. Create an object of Derived class and set parameters.

f. Create an object of CPEDAnalysisStatus and pass this object to it; calling its DoModal( ) method. This will start analysis.

g. On completion of analysis just call “UpdateAllViews”, this will update grid control with results.

4.7.2 Economic Dispatch Using GA

a. Derive a class from “CPEDGABase”.

b. Remaining steps are same as those in conventional approach.

4.7.3 Economic Dispatch Using Hybrid Approach

Sequential hybrid combinations have developed in such a way that control is transferred from one algorithm to another after specified number of iterations or convergence.

4.8 DOS Mode Operation

In the DOS mode all requisite data is read from text file and output is written on text files with following provisions:

- Output results with complete detail for every generation along with input data
- Output results of final generation with or without input.
- In case of multiple runs complete detail for every run with or without the summary of the final results of all the runs.
- In case of hybrid methodology complete output results for each generation for each algorithm or final results of each algorithm with or without input and summary of final results of all algorithms. Similar options exits for multiple runs of application.

Tracking of the time used in computation is also performed.
4.9 PED_Frame --- Execution Mode

The PED_Frame in Visual C environment appears as shown in Figures 4.2.

![Figure 4-2 PED_Frame in visual C++ Environment](image)

The File pull down menu as shown in Figure 4.3 is used for data read, save and print.

![Figure 4-3 File Pull Down Menu in PED_Frame](image)
The generators cost curves are loaded either from data file or may be entered through keyboard and appears in the PED_Frame as shown in Figure 4.4.

Figure 4-4 Cost curves of machines read from data file in PED_Frame Environment

The B-coefficients or loss coefficients are loaded from data file and shown in Figure 4.5.

Figure 4-5 B-coefficient read from data file in PED_Frame Environment
The Economic Dispatch menu gives the selection for different algorithms as shown in Figure 4.6.

![Image](image1.png)

**Figure 4-6 λ Iteration method in execution mode in PED_FRAME**

The parameters settings in PED_FRAME for economic dispatch using λ Iteration method are shown in Figure 4.7.

![Image](image2.png)

**Figure 4-7 Parameter settings for λ Iteration method in PED_FRAME**

The output of ED studies is shown in Figure 4.8. The output includes the power generation of the machines with their generation cost, total production cost, transmission loss and algorithm convergence parameters.
Figure 4-8: λ Iteration: Output results in PED_Frame

ED based on λ Iteration using GA has been selected as shown Figure 4.9 in PED_Frame environment including transmission loss.

Figure 4-9: λ Iteration using GA in PED_Frame
The GA parameters for ED studies are set in PED_Frame as given Figure 4.10.

![Figure 4-10 GA parameter settings in PED_Frame](image)

The progress during execution is given by progress indicator as shown in Figure (4.11).

![Figure 4-11 Progress indicator in PED_Frame](image)
Output in PED_Frame appears as shown in Figure (4.12) and (4.13).

![Figure 4-12: Iteration using GA: Output results in PED_Frame](image)

![Figure 4-13: Economic Dispatch using GA based on Real Power Search: Output results in PED_Frame](image)