CHAPTER 5

CONGESTION MANAGEMENT USING MULTI-OBJECTIVE OPTIMIZATION METHOD

Many real-world problems include simultaneous optimization of several objective functions. Generally, these functions are non-commensurable and often have competing and conflicting objectives. The presence of multiple objectives in a problem, in principle, gives rise to a set of optimal solutions (largely known as Pareto-optimal solutions). Multi-objective optimization with such conflicting objective functions leads to a set of optimal solutions, instead of one single optimal solution. The reason why many solutions are optimal is that no one can be considered to be better than any others regarding all objective functions. These optimal solutions are known as Pareto optimal solutions. A general multi-objective optimization problem consists of a number of objectives that should be optimized simultaneously associated with a number of equality and inequality constraints.

Congestion management methods available in the literature consider only one objective and provide only one single solution which does not provide any choice to the operator. Classical optimization methods (including the multi-criterion decision-making methods) suggest converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at a time. When such a method is to be used for finding multiple solutions, it has to be applied many times, for finding a different solution at each simulation run. The Non-Dominated Sorting Genetic Algorithm (NSGA) is one of the Pareto-based approaches for multi-objective optimization problems. Over the years, the main criticisms of the NSGA approach are:

(1) High computational complexity of non-dominated sorting,
(2) Lack of elitism and
(3) Need for specifying the sharing parameter.

All the above issues are sorted out using improved version of NSGA called NSGA II, that can find diverse set of solutions and converge near the true Pareto-optimal set. So, this efficient NSGA II algorithm is now proposed to solve the congestion management problem with two objectives considered simultaneously that can provide set of alternative solutions to the operator instead of single solution.

The general NSGA-II algorithm for multi-objective optimization problem is described below:

5.1 NSGA-II ALGORITHM

The various steps involved in the NSGA-II algorithm \[89\] are as follows:

1. Population Initialization

The population initialization is based on the problem range and constraints if any.

2. Non-Dominated sorting

The initialized population is sorted based on non-domination. The fast sort algorithm is described as below

- For each individual \( p \) in main population \( P \) do the following
  - Initialize \( S_p=\emptyset \). This set would contain all the individuals that is being dominated by \( p \).
  - Initialize \( n_p=0 \). This would be the number of individuals that dominate \( p \).
  - for each individual \( q \) in \( P \),
* if p dominated q then add q to the set $S_p$, i.e. $S_p = S_p \cup \{q\}$

* else if q dominates p then increment the domination counter for p. i.e. $n_p = n_p + 1$

- if $n_p = 0$ i.e. no individual dominates p then p belongs to first front; Set rank of the individual p to one i.e $p_{\text{rank}} = 1$. Update the first front set by add ind p to front one i.e $F_1 = F_1 \cup \{P\}$

  - This is carried out for all individuals in main population P.

  - Initialize the front counter to one. $i = 1$

  - Following is carried out while the $i^{th}$ front is nonempty i.e $F_i \neq \emptyset$

- $Q = \emptyset$. The set for sorting the individuals for $(i+1)^{th}$ front.

- for each individual p in front $F_i$

  * for each individual q in $S_p$ ( $S_p$ is the set of individuals dominated by p)

- $n_q = n_q - 1$, decrement the domination count for the individual q.

- if $n_q = 0$, then none of the individuals in the subsequent fronts would dominate q. Hence set $q_{\text{rank}} = i + 1$. Update the set Q with individual q i.e $Q = Q \cup q$.

- Increment the front counter by one

- Now the set Q is the next front and hence $F_i = Q$.

This algorithm is better than original NSGA since it utilizes the information about the set that an individual dominate ($S_p$) and number of individuals that dominate the individual ($n_p$).
3. Crowding distance

Once the non-dominated sort is complete the crowding distance is assigned. Since the individuals are selected based on the rank and crowding distance all the individuals in the population are assigned a crowding distance value. Crowding distance is assigned front wise and comparing the crowding distance between the two individuals in the different front is meaningless. The crowding distance is calculated as below

- For each front $F_i$, $n$ is the number of individuals
  - initialize the distance to be zero for all the individuals i.e $F_i(d_j)=0$; where $j^{th}$ individual in front $F_i$.
  - for each objective function $m$
    * Sort the individuals in front $F_i$ based on objective $m$. i.e. $I=\text{sort}(F_i, m)$
    * Assign infinite distance to boundary values for each individual in the $F_i$. i.e $I(d_1) = \infty$ and $I(d_n) = \infty$.
    * for $k=2$ to $(n-1)$, $I(d_k) = I(d_k) + \frac{I(k+1)m-1-I(k-1)m}{f_{m}^\text{max} - f_{m}^\text{min}}$ \hspace{1cm} (5.1)

$I(k)$ $m$ is the value of the $m^{th}$ objective function of the $k^{th}$ individual $I$.

The basic idea behind the crowding distance is the finding through Euclidian distance between the individual in a front based on their $m$ objectives in the $m$ dimensional hyper space. The individuals in the boundary are always selected since they have infinite distance assignment.
4. Selection

Once the individuals are sorted based on non-domination and crowding distance assigned, the selection is carried out using a crowded-comparison-operator \((\alpha_n)\). The comparison is carried out as below based on

1. non-domination rank \(p_{\text{rank}}\) i.e. individuals in front \(F_i\) will have their rank as \(p_{\text{rank}} = i\).

2. Crowding distance \(F_i(d_i)\)
   - \(p\), \(q\) \(\alpha_n\) q if
     - \(p_{\text{rank}} < q_{\text{rank}}\)
     - or if \(p\) and \(q\) belong to the same front \(F_i\) then \(F_i(d_p) > F_i(d_q)\) i.e. the crowding distance should be more.

The individuals are selected by using a binary tournament selection with crowded-comparison-operator.

5. Genetic Operators

Real coded GA uses Simulated Binary Crossover (SBX) operator for crossover and polynomial mutation.

5.A Simulated Binary Crossover (SBX)

Simulated binary crossover simulates the binary crossover observed in the nature and is give as below.

\[
c_{1,k} = \frac{1}{2} \left[ (1 - \beta_k) p_{1,k} + (1 + \beta_k) p_{2,k} \right] \quad (5.2)
\]

\[
c_{2,k} = \frac{1}{2} \left[ (1 + \beta_k) p_{1,k} + (1 - \beta_k) p_{2,k} \right] \quad (5.3)
\]
where $c_{i,k}$ is the $i^{th}$ child with $k^{th}$ parent $p_{i,k}$ is the selected parent and $\beta_k$ ($\geq 0$) is a sample from a random number generated having the density

\[
p(\beta) = \frac{1}{2} (\eta_c + 1) \beta^{\eta_c}, \text{ if } 0 \leq \beta \leq 1
\]

\[
p(\beta) = \frac{1}{2} (\eta_c + 1) \frac{1}{\beta^{\eta_c+1}}, \text{ if } \beta > 1
\]

This distribution can be obtained from a uniformly sampled random number $u$ between $(0,1)$. $\eta_c$ is the distribution index for crossover. That is

\[
\beta(u) = (2u)^{\eta_c+1}
\]

\[
\beta(u) = \frac{1}{2(1-u)^{\eta_c+1}}
\]

### 5.B Polynomial Mutation

\[
c_k = p_k + (p_k^u - p_k^l) \delta_k
\]

where $c_k$ is the child and $p_k$ is the parent with $p_k^u$ being the upper bound on the parent component, $p_k^l$ is the lower bound and $\delta_k$ is small variation which is calculated from a polynomial distribution by using

\[
\delta_k = (2r_k)^{\eta_m-1} - 1, \text{ if } r_k < 0.5
\]

\[
\delta_k = 1 - [2(1-r_k)]^{\eta_m+1}, \text{ if } r_k \geq 0.5
\]

$r_k$ is an uniformly sampled random number between $(0,1)$ and $\eta_m$ is the mutation distribution index.

### 6. Recombination and Selection

The offspring population is combined with the current generation population and the selection is performed to set the individuals of the next
generation. Since all the previous and current best individuals are added in the population, elitism is ensured. Population is now sorted based on the non-domination. The new generation is filled by each front subsequently until the population size exceeds the current population size. If by adding all the individuals in front $F_j$ the population exceeds $N$ then individuals in front $F_j$ are selected based on their crowding distance in the descending order until population size is $N$. This process is repeated to generate the subsequent generations.

This completes one generation of the NSGA-II algorithm. The stopping criteria would be the maximum number of generations.

So far, in the earlier chapters congestion management problem with single objective is considered. In this chapter multi-objective congestion management problem is addressed. In this study, the following two types of objective works are considered in the congestion management problem.

1. Congestion management for minimizing the congestion cost and line losses by generation rescheduling and/or load shedding.

2. Congestion management for minimizing the cost of installation of DGs and the line losses by optimal location and sizing of DGs.

The objectives considered and the step-by-step procedure for solving the above mentioned problems are discussed in the subsequent sections.

5.2 MINIMIZING CONGESTION COST AND LINE LOSS BY GENERATOR RESCHEDULING AND LOAD SHEDDING

The shift in active power generations and/or demands to minimize the total congestion cost and line loss as a multi-objective optimization problem is given as follows

Objective function 1

\[
TC = \sum_{j=1}^{Ng} \left( C_{Gj}^{u} \Delta P_{Gj}^{u} + C_{Gj}^{d} \Delta P_{Gj}^{d} \right) + \sum_{i=1}^{Nd} \left( C_{Di}^{d} \Delta P_{Di}^{d} \right) + \alpha \times CI
\]  

(5.11)
where \( \alpha = \) weighting factor

\[
CI = \sum_{ij}^{NL} \left( P_{ij} - P_{ij}^{\text{max}} \right)^2
\]

Objective function 2

\[
P = \sum_{L=1}^{NL} (P_{ij} + P_{ji})
\] (5.12)

where \( L \) is the line connecting buses \( i \) and \( j \) and \( P_{ij} \) and \( P_{ji} \) are the real power flows from bus \( i \) to \( j \) and from bus \( j \) to \( i \) respectively and \( P \) is the total real power losses.

The constraints considered are (2.7) to (2.18) as discussed in Chapter 2.

### 5.3 MINIMIZING DG COST AND LINE LOSS BY OPTIMAL SIZING AND PLACEMENT OF DG

The optimal location and sizing of DG to minimize the DG cost and line loss as a multi-objective optimization problem is given as follows

Objective function 1

\[
C(P_{DG}) = a_{DG} + b_{DG}P_{DG} + c_{DG}(P_{DG})^2
\] (5.13)

Objective function 2

\[
P = \sum_{L=1}^{NL} (P_{ij} + P_{ji})
\] (5.14)

where \( P_{ij} \) and \( P_{ji} \) are the real power flows from bus \( i \) to \( j \) and from bus \( j \) to \( i \) respectively and \( P \) is the total real power losses in the system.

The constraints are (2.7) to (2.18) as discussed in Chapter 2.

Additional constraint of

\[
P_{DG}^{\text{min}} \leq P_{DG} \leq P_{DG}^{\text{max}}
\] (5.15)

must also be satisfied for the DG
5.4 MULTI-OBJECTIVE OPTIMIZATION ALGORITHM FOR GENERATOR RESCHEDULING AND LOAD SHEDDING

The step-by-step procedure for finding the pareto-optimal solutions for minimizing the congestion cost and line losses by generation and/or load shedding is given below.

1. Set up NSGA II parameters like population size, number of generations, distribution indices for crossover (mu), and mutation (mum). Here both mu and mum are taken as 20 each.

2. Read line data, bus data, incremental and decrement bidding costs for each generator. When applying evolutionary computation algorithm, the first step is to decide the control variables embedded in the individuals. In this work, control variables are the generator real power re-dispatches. The control variables are generated randomly satisfying their practical operation constraints.

3. For each chromosome of population, calculate the values of objective function-1 using Equation (5.11) and objective function-2 using Equation (5.12).

4. The equality and inequality constraints are handled by Newton-Raphson Power Flow.

5. Non-domination sorting of population is carried out. Tournament selection is then applied to select the best individuals based on the crowding distance.

6. Crossover and Mutation operators are carried out to generate offspring (Qt) and the new vectors obtained must satisfy the limits; if not, set it to the appropriate extrema.

7. Calculate the value of each objective function of Qt and merge the parent and offspring population to preserve elites.
(8) Again perform non-dominated sorting on the combined population based on the crowding distance measure and obtain the best new parent population (Pt +1) of size N out of 2N population, and this would be the parents for next generation and this process is carried out till the maximum number of generations is reached.

(9) Finally pareto front is achieved, that is, a set of solutions satisfying both objectives are obtained.

5.5 MULTIOBJECTIVE OPTIMIZATION ALGORITHM FOR OPTIMAL LOCATION AND SIZING OF DG

The step-by-step procedure for finding the pareto-optimal solutions for minimizing the DG cost and line losses by optimal location and sizing of DG is given below:

(1) Set up NSGA II parameters like population size, number of generations, distribution indices for crossover (mu), and mutation (mum). Here also both mu and mum are taken as 20.

(2) Read line data, bus data, incremental and decrement bidding costs for each generator. When applying evolutionary computation algorithm, the first step is to decide the control variables embedded in the individuals. The control variables, location and size of DG are generated randomly satisfying their practical operation constraints.

(3) For each chromosome of population, calculate the values of objective function-1 using Equation (5.13) and objective function-2 using Equation (5.14).

(4) The equality and inequality constraints are handled by Newton-Raphson Power Flow.
(5) Non-domination sorting of population is carried out. Tournament selection is then applied to select the best individuals based on the crowding distance.

(6) Crossover and Mutation operators are carried out to generate offspring (Qt) and the new vectors obtained must satisfy the limits; if not, set it to the appropriate extrema.

(7) Calculate the value of each objective function of Qt and merge the parent and offspring population to preserve elites.

(8) Again perform non-dominated sorting on the combined population based on the crowding distance measure and obtain the best new parent population (Pt +1) of size N out of 2N population, so this would be the parents for next generation and this process is carried out till the maximum number of generations is reached.

(9) Finally pareto front, a set of solutions satisfying both objectives are obtained.

5.6 RESULTS AND DISCUSSION

The proposed algorithm is implemented and tested on IEEE 14 and 30 bus systems for the similar cases considered in the earlier chapters.

The set of pareto-optimal solution for all the three cases considered in IEEE 14 bus system for minimizing the congestion cost (objective function 1) and line losses (objective function 2) by rescheduling the generators and / or load are given in Figures 5.1-5.3. Three non-dominated solutions from the truncated archive are presented in Table 5.1.

The set of pareto-optimal solution for all the three cases considered in IEEE 14 bus system for minimizing the DG cost (objective function 1) and line losses (objective function 2) by optimal location and sizing of DG are given in Figures 5.4 - 5.6. Three non-dominated solutions from the truncated archive are presented in Table 5.2.
Similar results are obtained for IEEE 30 bus system and the results are presented in the Figures 5.7-5.12 and in Tables 5.3 and 5.4.

Figure 5.1 Pareto front for Case 1A by generator rescheduling and load shedding

Figure 5.2 Pareto front for Case 1B by generator rescheduling and load shedding
Figure 5.3 Pareto front for Case 1C by generator rescheduling and load shedding

Table 5.1 Three solutions among the pareto front for generator rescheduling and load shedding for IEEE 14 Bus system

<table>
<thead>
<tr>
<th>Cases</th>
<th>Congestion Cost ($/hr)</th>
<th>Real Power Loss (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>1041</td>
<td>73.11</td>
</tr>
<tr>
<td></td>
<td>2550</td>
<td>72.19</td>
</tr>
<tr>
<td></td>
<td>3688</td>
<td>72.05</td>
</tr>
<tr>
<td>1B</td>
<td>3953</td>
<td>72.93</td>
</tr>
<tr>
<td></td>
<td>4988</td>
<td>72.20</td>
</tr>
<tr>
<td></td>
<td>6251</td>
<td>71.82</td>
</tr>
<tr>
<td>1C</td>
<td>3202</td>
<td>75.61</td>
</tr>
<tr>
<td></td>
<td>3594</td>
<td>74.88</td>
</tr>
<tr>
<td></td>
<td>4071</td>
<td>74.17</td>
</tr>
</tbody>
</table>
Figure 5.4 Pareto front for Case 1A by optimal location and sizing of DG

Figure 5.5 Pareto front for Case 1B by optimal location and sizing of DG
Figure 5.6 Pareto front for Case 1C by optimal location and sizing of DG

Table 5.2 Three solutions among the pareto front for optimal location and sizing of DG for IEEE 14 Bus system

<table>
<thead>
<tr>
<th>Cases</th>
<th>DG Cost ($/hr)</th>
<th>Real power Loss (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>2327.62</td>
<td>73.424</td>
</tr>
<tr>
<td></td>
<td>2312.04</td>
<td>73.425</td>
</tr>
<tr>
<td></td>
<td>2280.99</td>
<td>73.429</td>
</tr>
<tr>
<td>1B</td>
<td>2350.72</td>
<td>72.37</td>
</tr>
<tr>
<td></td>
<td>2234.88</td>
<td>72.45</td>
</tr>
<tr>
<td></td>
<td>2166.62</td>
<td>72.57</td>
</tr>
<tr>
<td>1C</td>
<td>2514.27</td>
<td>74.598</td>
</tr>
<tr>
<td></td>
<td>2476.28</td>
<td>74.601</td>
</tr>
<tr>
<td></td>
<td>2456.46</td>
<td>74.605</td>
</tr>
</tbody>
</table>
Figure 5.7 Pareto front for Case 2A by generator rescheduling and load shedding

Figure 5.8 Pareto front for Case 2B by generator rescheduling and load shedding
Figure 5.9 Pareto front for Case 2C by generator rescheduling and load shedding

Table 5.3 Three solutions among the pareto front for generator rescheduling and load shedding for IEEE 30 Bus system

<table>
<thead>
<tr>
<th>Cases</th>
<th>Congestion Cost ($/hr)</th>
<th>Real Power Loss (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>597</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>2090</td>
<td>6.31</td>
</tr>
<tr>
<td></td>
<td>6490</td>
<td>2.54</td>
</tr>
<tr>
<td>2B</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1230</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>1866</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td>2897</td>
<td>4.7</td>
</tr>
<tr>
<td>2C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2051</td>
<td>9.5</td>
</tr>
<tr>
<td></td>
<td>3420</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>4759</td>
<td>5.4</td>
</tr>
</tbody>
</table>
Figure 5.10 Pareto front for Case 2A by optimal location and sizing of DG

Figure 5.11 Pareto front for Case 2B by optimal location and sizing of DG
Figure 5.12 Pareto front for Case 2C by optimal location and sizing of DG

Table 5.4 Three solutions among the pareto front for optimal location and sizing of DG for IEEE 30 Bus system

<table>
<thead>
<tr>
<th>Cases</th>
<th>Congestion Cost ($/hr)</th>
<th>Real Power Loss (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2A</td>
<td>1024</td>
<td>16.29</td>
</tr>
<tr>
<td></td>
<td>1524</td>
<td>8.69</td>
</tr>
<tr>
<td></td>
<td>2193</td>
<td>5.99</td>
</tr>
<tr>
<td>2B</td>
<td>1838.45</td>
<td>14.84</td>
</tr>
<tr>
<td></td>
<td>2060.32</td>
<td>12.75</td>
</tr>
<tr>
<td></td>
<td>2247.75</td>
<td>11.44</td>
</tr>
<tr>
<td>2C</td>
<td>2302</td>
<td>13.42</td>
</tr>
<tr>
<td></td>
<td>2783</td>
<td>12.01</td>
</tr>
<tr>
<td></td>
<td>3143</td>
<td>9.51</td>
</tr>
</tbody>
</table>
5.7 CONCLUSION

When multi-objective problem is converted into single objective optimization problem, only one compromised solution considering both the objectives is obtained and this does not provide any choice to the operators. When this method is used to find multiple solutions, it has to be run many times hopefully for finding a different solution in each simulation run. So it is time consuming and cannot be used for large practical systems. In the method presented in this chapter, both the objectives are considered separately and solved using multi-objective, NSGA II method. Though multi-objective GA does not guarantee global optimal solutions it provides solution very close to the sub-optimal solutions. This method provides a set of pareto-optimal solutions for any congestion problem, giving the system operator an option for judicious decision in solving congestion.

The multi-objective congestion management method discussed in this chapter gives a set of pareto solutions in one simulation run by either generation rescheduling or optimal sizing and location of DGs. The rescheduling of the generators for effective congestion management is a tedious and time consuming process in real time complex system. Hence a simpler congestion management method, based on congestion zones/ clusters, is presented in the next chapter.