CONDITION OF SUBSTANCE SEARCH ALGORITHM FOR SOLVING REACTIVE POWER PROBLEM

Dr. K. Lenin *1

*1 Professor, Department of EEE, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada, Andhra Pradesh -520007, India

Abstract

In this paper, Condition of Substance Search (COS) algorithm is introduced to solve optimal reactive power dispatch problem. The Condition of Substance Search (COS) algorithm is based on the simulation of the shape of substance incidence. In Condition of Substance Search (COS) algorithm, individuals follow molecules which interrelate to each other by using evolutionary operations which are based on the corporal principles of the thermal-energy motion mechanism. The algorithm is developed by considering each condition of substance in harmony with different exploration–exploitation ratio. The evolutionary progression is alienated into three phases which emulate the three condition of substance: solid, liquid & gas. This technique can considerably improve the equilibrium between exploration–exploitation, however preserving the high-quality search ability of an evolutionary approach. The proposed Condition of Substance Search (COS) algorithm has been tested on standard IEEE 30 bus test system and simulation results show clearly the improved performance of the projected Condition of Substance Search (COS) algorithm in reducing the real power loss and voltage stability also enhanced.

Keywords: Condition of Substance Search; Exploration & Exploitation; Optimal Reactive Power; Transmission Loss.


1. Introduction

In recent years the optimal reactive power dispatch (ORPD) problem has received great attention as a result of the improvement on economy and security of power system operation. Solutions of ORPD problem aim to minimize object functions such as fuel cost, power system loses, etc. while satisfying a number of constraints like limits of bus voltages, tap settings of transformers, reactive and active power of power resources and transmission lines and a number of controllable Variables. Various numerical methods like the gradient method [1-2], Newton method [3] and linear programming [4-7] have been implemented to solve the optimal reactive power dispatch problem. Both the gradient and Newton methods have the intricacy in managing
inequality constraints. The problem of voltage stability and collapse play a key role in power system planning and operation [8]. Evolutionary algorithms such as genetic algorithm have been already projected to solve the reactive power flow problem [9-11]. Evolutionary algorithm is a heuristic methodology used for minimization problems by utilizing nonlinear and non-differentiable continuous space functions. In [12], Hybrid differential evolution algorithm is projected to increase the voltage stability index. In [13] Biogeography Based algorithm is projected to solve the reactive power dispatch problem. In [14], a fuzzy based method is used to solve the optimal reactive power scheduling method. In [15], an improved evolutionary programming is used to elucidate the optimal reactive power dispatch problem. In [16], the optimal reactive power flow problem is solved by integrating a genetic algorithm with a nonlinear interior point method. In [17], a pattern algorithm is used to solve ac-dc optimal reactive power flow model with the generator capability limits. In [18], F. Capitanescu proposes a two-step approach to calculate Reactive power reserves with respect to operating constraints and voltage stability. In [19], a programming based approach is used to solve the optimal reactive power dispatch problem. In [20], A. Kargarian et al present a probabilistic algorithm for optimal reactive power provision in hybrid electricity markets with uncertain loads. This paper put forward Condition of Substance Search (COS) algorithm to solve reactive power dispatch problem. This algorithm is developed by considering each condition of substance at different exploration–exploitation ratio. Consequently, the evolutionary progression is divided into three phases which imitate the three conditions of matter: gas, liquid and solid. At every condition, molecules (individuals) reveal dissimilar behaviours [21]. Beginning from the gas state (wholesome exploration), the algorithm modify the deliberation of exploration and exploitation until the solid state (wholesome exploitation) is reached. As a result, the process can significantly step forward to balance between exploration–exploitation, however preserving the high-quality exploration ability of an evolutionary approach. Proposed Condition of Substance Search (COS) algorithm has been evaluated on standard IEEE 30 bus test system. The simulation results show that our proposed approach outperforms all the reported standard algorithms in minimization of real power loss & voltage stability enhanced.

2. Voltage Stability Evaluation

2.1. Modal analysis for Voltage Stability Evaluation

Modal analysis is one among best methods for voltage stability enhancement in power systems. The steady state system power flow equations are given by,

\[
\begin{bmatrix}
\Delta P \\
\Delta Q \\
\Delta \theta \\
\Delta V
\end{bmatrix} = 
\begin{bmatrix}
J_{p\theta} & J_{pV} \\
J_{q\theta} & J_{qV}
\end{bmatrix}
\begin{bmatrix}
\Delta \theta \\
\Delta V
\end{bmatrix}
\]

Where
\(\Delta P\) = Incremental change in bus real power.
\(\Delta Q\) = Incremental change in bus reactive Power injection
\(\Delta \theta\) = incremental change in bus voltage angle.
\(\Delta V\) = Incremental change in bus voltage Magnitude.

\(J_{p\theta}, J_{pV}, J_{q\theta}, J_{qV}\) jacobian matrix are the sub-matrixes of the System voltage stability is affected by both P and Q.
To reduce (1), let $\Delta P = 0$, then.
$$\Delta Q = [J_{QV} - J_{Q\theta}J_{P\theta}^{-1}J_{PV}]\Delta V = J_R\Delta V$$
$$\Delta V = J^{-1}\Delta Q$$

Where
$$J_R = (J_{QV} - J_{Q\theta}J_{P\theta}^{-1}J_{PV})$$
$J_R$ is called the reduced Jacobian matrix of the system.

### 2.2. Modes of Voltage Instability

Voltage Stability characteristics of the system have been identified by computing the Eigen values and Eigen vectors.

Let $$J_R = \xi \Lambda \eta$$

Where,
- $\xi$ = right eigenvector matrix of $J_R$
- $\eta$ = left eigenvector matrix of $J_R$
- $\Lambda$ = diagonal eigenvalue matrix of $J_R$ and

$$J_R^{-1} = \xi \Lambda^{-1} \eta$$

From (5) and (8), we have
$$\Delta V = \xi \Lambda^{-1} \eta \Delta Q$$

or
$$\Delta V = \sum_i \frac{\xi_i \eta_i}{\lambda_i} \Delta Q$$

Where $\xi_i$ is the $i$th column right eigenvector and $\eta$ the $i$th row left eigenvector of $J_R$.
$\lambda_i$ is the $i$th Eigen value of $J_R$.

The $i$th modal reactive power variation is,
$$\Delta Q_{mi} = K_i \xi_i$$

where,
$$K_i = \sum_j \xi_{ij}^2 - 1$$

Where
$\xi_{ji}$ is the $j$th element of $\xi_i$

The corresponding $i$th modal voltage variation is
$$\Delta V_{mi} = [1/\lambda_i] \Delta Q_{mi}$$

If $|\lambda_i| = 0$ then the $i$th modal voltage will collapse.

In (10), let $\Delta Q = e_k$ where $e_k$ has all its elements zero except the $k$th one being 1. Then,
$$\Delta V = \sum_i \eta_{ik} \xi_i \frac{\xi_k}{\lambda_i}$$
$$\eta_{1k}$ k th element of $\eta_1$
3. Problem Formulation

The objectives of the reactive power dispatch problem is to minimize the system real power loss and maximize the static voltage stability margins (SVSM).

3.1. Minimization of Real Power Loss

Minimization of the real power loss (Ploss) in transmission lines is mathematically stated as follows.

\[
\text{P}_{\text{loss}} = \sum_{k=1}^{n} g_k (v_i^2 + v_j^2 - 2v_i v_j \cos \theta_{ij})
\]

(14)

Where \( n \) is the number of transmission lines, \( g_k \) is the conductance of branch \( k \), \( v_i \) and \( v_j \) are voltage magnitude at bus \( i \) and bus \( j \), and \( \theta_{ij} \) is the voltage angle difference between bus \( i \) and bus \( j \).

3.2. Minimization of Voltage Deviation

Minimization of the voltage deviation magnitudes (VD) at load buses is mathematically stated as follows.

\[
\text{Minimize VD} = \sum_{k=1}^{nl} |v_k - 1.0|
\]

(15)

Where \( nl \) is the number of load busses and \( v_k \) is the voltage magnitude at bus \( k \).

3.3. System Constraints

Objective functions are subjected to these constraints shown below.

Load flow equality constraints:

\[
P_{G_i} - P_{D_i} - \sum_{j=1}^{nb} v_i (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0, i = 1, 2, ..., nb
\]

(16)

\[
Q_{G_i} - Q_{D_i} - \sum_{j=1}^{nb} v_j (G_{ij} \sin \theta_{ij} + B_{ij} \cos \theta_{ij}) = 0, i = 1, 2, ..., nb
\]

(17)

where, \( nb \) is the number of buses, \( P_G \) and \( Q_G \) are the real and reactive power of the generator, \( P_D \) and \( Q_D \) are the real and reactive load of the generator, and \( G_{ij} \) and \( B_{ij} \) are the mutual conductance and susceptance between bus \( i \) and bus \( j \).

Generator bus voltage (\( V_{Gi} \)) inequality constraint:

\[
V_{Gi}^{\min} \leq V_{Gi} \leq V_{Gi}^{\max}, i \in ng
\]

(18)
Load bus voltage (VLi) inequality constraint:
\[ V_{Li}^{\text{min}} \leq V_{Li} \leq V_{Li}^{\text{max}}, \; i \in n_l \]  
(19)

Switchable reactive power compensations (QCi) inequality constraint:
\[ Q_{Ci}^{\text{min}} \leq Q_{Ci} \leq Q_{Ci}^{\text{max}}, \; i \in n_c \]  
(20)

Reactive power generation (QGi) inequality constraint:
\[ Q_{Gi}^{\text{min}} \leq Q_{Gi} \leq Q_{Gi}^{\text{max}}, \; i \in n_g \]  
(21)

Transformers tap setting (Ti) inequality constraint:
\[ T_i^{\text{min}} \leq T_i \leq T_i^{\text{max}}, \; i \in n_t \]  
(22)

Transmission line flow (SLi) inequality constraint:
\[ S_{Li}^{\text{min}} \leq S_{Li} \leq S_{Li}^{\text{max}}, \; i \in n_l \]  
(23)

Where, nc, ng and nt are numbers of the switchable reactive power sources, generators and transformers

4. Condition of Substance

The substance can take different phases which are commonly known as states. Traditionally, three condition of substance are known: solid, liquid, and gas. The differences among such states are based on forces which are exerted among particles composing a material [22].

![Figure 1(a):](image)

![Figure 1(b):](image)
In the gas phase, molecules present enough kinetic energy so that the effect of intermolecular forces is small (or zero for an ideal gas), while the typical distance between neighbouring molecules is greater than the molecular size. A gas has no definite shape or volume, but occupies the entire container in which it is confined. Fig. 1(a) [21] shows the movements exerted by particles in a gas state. The movement experimented by the molecules represent the maximum permissible displacement $\rho_1$ among particles [23]. In a liquid state, intermolecular forces are more restrictive than those in the gas state. The molecules have enough energy to move relatively to each other still keeping a mobile structure. Therefore, the shape of a liquid is not definite but is determined by its container. Fig. 1(b) [21] presents a particle movement $\rho_2$ within a liquid state. Such movement is smaller than those considered by the gas state but larger than the solid state [24]. In the solid state, particles (or molecules) are packed together closely with forces among particles being strong enough so that the particles cannot move freely but only vibrate. As a result, a solid has a stable, definite shape and a definite volume. Solids can only change their shape by force, as when they are broken or cut. Fig. 1(c) [21] shows a molecule configuration in a solid state. Under such conditions, particles are able to vibrate considering a minimal $\rho_3$ distance [23].

5. Condition of Substance Search (COS) algorithm

A. Explanation of Functioning

In the approach, individuals are considered as molecules whose positions on a multidimensional space are modified as the algorithm progresses. The movement of such molecules is driven by the analogy to the motion of thermal-energy. The velocity and direction of each molecule’s movement are determined by considering the collision, the attraction forces and the arbitrary phenomena experimented by the molecule set [25]. In our method, such behaviours have been applied by defining several operators such as the direction vector, the collision and the random positions operators, all of which emulate the behaviour of actual physics laws. The direction vector operator allocates a direction to each molecule in order to lead the particle movement as the evolution procedure takes place. On the other side, the collision operator mimics those collisions that are experimented by molecules as they interact to each other. A collision is considered when the distance between two molecules is shorter than a determined proximity distance. The collision operator is thus implemented by interchanging directions of the involved molecules. In order to simulate the arbitrary behaviour of molecules, the proposed algorithm...
produces arbitrary positions following a probabilistic condition that considers arbitrary locations within a feasible exploration space.

**B. Path Vector**

The direction vector operator mimics the way in which molecules change their positions as the evolution process develops. For each n-dimensional molecule $P_i$ from the population $P$, it is assigned an n dimensional direction vector $d_i$ which stores the vector that controls the particle movement. Initially, all the direction vectors $\left(D = \{d_1, d_2, \ldots, d_{N_p}\}\right)$ are randomly chosen within the range of [-1, 1]. As the system evolves, molecules experiment several attraction forces. In order to simulate such forces, the proposed algorithm implements the attraction phenomenon by moving each molecule towards the best so-far particle. Therefore, the new direction vector for each molecule is iteratively computed considering the following model:

$$d_{i}^{k+1} = d_{i}^{k} \cdot \left(1 - \frac{k}{\text{gen}}\right) \cdot 0.5 + a_i$$

(24)

where $a_i$ represents the attraction unitary vector calculated as $a_i = (p_{best} - P_i)/\|p_{best} - P_i\|$, being $p_{best}$ the best individual seen so-far, while $P_i$ is the molecule $i$ of population $P$. $k$ represents the iteration number, whereas $\text{gen}$ involves the total iteration number that constitutes the complete evolution process. Under this operation, each particle is moved towards a new direction which combines the past direction, which was initially computed, with the attraction vector over the best individual seen so-far. It is important to point out that the relative importance of the past direction decreases as the evolving process advances.

In order to calculate the new molecule position, it is necessary to compute the velocity $V_i$ of each molecule by using:

$$V_i = d_i \cdot v_{init}$$

(25)

Being $v_{init}$ the initial velocity magnitude which is calculated as follows:

$$v_{init} = \frac{\sum_{j=1}^{n}(b_{j,high} - b_{j,low})}{n} \cdot \beta$$

(26)

Where $b_{j,low}$ and $b_{j,high}$ are the low $j$ parameter bound and the upper $j$ parameter bound respectively, whereas $\beta \in [0,1]$.

Then, the new position for each molecule is updated by:

$$P_{i,j}^{k+1} = P_{i,j}^{k} + v_{i,j} \cdot \text{rand}(0,1) \cdot \rho \cdot (b_{j,high} - b_{j,low})$$

(27)

Where $0.5 \leq \rho \leq 1$.

**C. Collision**

The collision operator mimics the collisions experimented by molecules while they interact to each other. Collisions are calculated if the distance between two molecules is shorter than a determined proximity value. Therefore, if $\|P_i - P_q\| < r$, a collision between molecules $i$ and $q$ is assumed; otherwise, there is no collision, considering $i, q \in \{1, \ldots, N_p\}$ such that $i \neq q$. If a
collision occurs, the direction vector for each particle is modified by interchanging their respective direction vectors as follows:
\[ d_i = d_q \text{ and } d_q = d_i \]  
(28)

The collision radius is calculated by:
\[ r = \frac{\sum_{j=1}^{n} (b_{j}^{high} - b_{j}^{low})}{n} \cdot \alpha \]  
(29)

Where \( \alpha \in [0,1] \)

Under this operator, a spatial region enclosed within the radius \( r \) is assigned to each particle. In case the particle regions collide to each other, the collision operator acts upon particles by forcing them out of the region. The radio \( r \) and the collision operator provide the ability to control diversity throughout the search process. In other words, the rate of increase or decrease of diversity is predetermined for each stage. The collision incorporation therefore enhances the exploratory behaviour in the proposed approach.

**D. Arbitrary Location**

In order to simulate the random behaviour of molecules, the proposed algorithm generates random positions following a probabilistic criterion within a feasible search space. For this operation, a uniform random number \( r_m \) is generated within the range \([0, 1]\). If \( r_m \) is smaller than a threshold \( H \), a random molecule’s position is generated; otherwise, the element remains with no change. Therefore such operation can be modelled as follows:
\[
P_{i,j}^{k+1} = \begin{cases} 
    b_{j}^{low} + \text{rand}(0,1) \cdot (b_{j}^{high} - b_{j}^{low}) & \text{with probability } H \\
    P_{i,j}^{k+1} & \text{with porbability } 1 - H 
\end{cases} 
\]  
(30)

Where \( i \in \{1,..,N_p\} \) and \( j \in \{1,..,n\} \)

**E. Most Excellent Component Modernizing**

Despite this updating operator does not belong to condition of substance metaphor, it is used to simply store the best so-far solution. In order to update the best molecule \( p^{\text{best}} \) seen so-far, the best found individual from the current k population \( p_{\text{best},k} \) is compared to the best individual \( p_{\text{best},k-1} \) of the last generation. If \( p_{\text{best},k} \) is better than \( p_{\text{best},k-1} \) according to its fitness value, best \( p \) is updated with \( p_{\text{best},k} \), otherwise \( p_{\text{best}} \) remains with no change. Therefore, \( p_{\text{best}} \) stores the best historical individual found so-far.

**F. Common Process**

At each stage, the same operations are implemented. However, depending on which state is referred, they are employed considering a different parameter configuration. Such procedure is composed by five steps and maps the current population \( P^k \) to a new population \( P^{k+1} \). The algorithm receives as input the current population \( P^k \) and the configuration parameters \( \rho, \beta, \alpha \) and \( H \), where as it yields the new population \( P^{k+1} \).

Step a: Find the best element of the population \( p^{\text{best}} \in \{P\} \)
Step b: Calculate \( v_{init} \) and \( r \)
Step c: Compute the new molecules by using the Direction vector operator.
Step d: Solve collisions by using the Collision operator
Step e: Generate new random positions by using the Random positions operator

G. The Comprehensive Algorithm

The comprehensive algorithm is divided into four different parts. The first corresponds to the initialization stage, whereas the last three represent the condition of substance. All the optimization process, which consists of a gen number of iterations, is organized into three different asymmetric phases, employing 50% of alliterations for the gas state (exploration), 40% for the liquid state (exploration-exploitation) and 10% for the solid state (exploitation).

Initialization

The algorithm begins by initializing a set \( P \) of \( N_p \) molecules (\( P = \{P_1, P_2, \ldots, P_{N_P}\} \)) each molecule position \( p_i \) is an \( n \)-dimensional vector containing the parameter values to be optimized. Such values are randomly and uniformly distributed between the pre-specified lower initial parameter bound \( b_j^{\text{low}} \) and the upper initial parameter bound \( b_j^{\text{high}} \), just as it is described by the following expressions:

\[
P_{i,j}^0 = b_j^{\text{low}} + \text{rand}(0,1) \cdot (b_j^{\text{high}} - b_j^{\text{low}})
\]

\( j=1,2,\ldots,N, \ i=1,2,\ldots,N_p \)

Where \( j \) and \( i \) are the parameter and molecule index respectively whereas zero indicates the initial population. Hence, \( P_{i}^j \) is the \( j \)-th parameter of the \( i \)-th molecule.

Gas Condition

In the gas state, molecules experiment severe displacements and collisions. Such state is characterized by random movements produced by non-modelled molecule phenomena. Therefore, the \( \rho \) value from the direction vector operator is set to a value near to one so that the molecules can travel longer distances. Similarly, the \( H \) value representing the random positions operator is also configured to a value around one, in order to allow the random generation for other molecule positions. The gas state is the first phase and lasts for the 50% of all iterations which compose the complete optimization process. The computational procedure for the gas state can be summarized as follows:

Step a: Set the parameters \( \rho \in [0.8, 1], \beta = 0.8, \alpha = 0.8 \text{ and } H=0.9 \) being consistent with the gas state.

Step b: Apply the general procedure

Step c: If the 50% of the total iteration number is completed \( (1 \leq k \leq 0.5 \leq \text{gen}) \), then the process continues to the liquid state procedure; otherwise go back to step b.

Liquid Condition

Although molecules currently at the liquid state exhibit restricted motion in comparison to the gas state, they still show a higher flexibility with respect to the solid state. Furthermore, the generation of random positions which are produced by non-modelled molecule phenomena is scarce. For this reason, the \( \rho \) value from the direction vector operator is bounded to a value between 0.30 and 0.60. Similarly, the random position operator \( H \) is configured to a value near to cero in order to allow the random generation of fewer molecule positions. In the liquid state, collisions are also less common than in gas state, so the collision radius, which is controlled by,
is set to a smaller value in comparison to the gas state. The liquid state is the second phase and lasts the 40% of all iterations which compose the complete optimization process. The computational procedure for the liquid state can be summarized as follows:
Step d: Set the parameters \( \rho \in [0.3, 0.6] \), \( \beta = 0.4 \), \( \alpha = 0.2 \) and \( H=0.2 \) being consistent with the liquid state.
Step e: Apply the general procedure.
Step f: If the 90% (50% from the gas state and 40% from the liquid state) of the total iteration number is completed \( (0.5 \cdot gen < k \leq 0.9 \cdot gen) \), then the process continues to the solid state procedure; otherwise go back to step e.

**Solid Condition**

In the solid state, forces among particles are stronger so that particles cannot move freely but only vibrate. As a result, effects such as collision and generation of random positions are not considered. Therefore the \( \rho \) value of the direction vector operator is set to a value near to zero indicating that the molecules can only vibrate around their original positions. The solid state is the third phase and lasts for the 10% of all iterations which compose the complete optimization process. The computational procedure for the solid state can be summarized as follows:
Step g: Set the parameters \( \rho \in [0.0, 0.1] \) and \( \beta= 0.1, \alpha = 0 \) and \( H=0 \) being consistent with the solid state.
Step h: Apply common process.
Step i: If the 100% of the total iteration number is completed \( (0.9 \cdot gen < k \leq gen) \), the process is finished; otherwise go back to step h.

It is important to clarify that the use of this particular configuration \( (\alpha = 0 \) and \( H=0) \) disables the collision and generation of random positions operators which have been illustrated in the general procedure.

### 6. Simulation Results

The efficiency of the proposed Condition of Substance Search (COS) algorithm is demonstrated by testing it on standard IEEE-30 bus system. The IEEE-30 bus system has 6 generator buses, 24 load buses and 41 transmission lines of which four branches are (6-9), (6-10), (4-12) and (28-27) - are with the tap setting transformers. The lower voltage magnitude limits at all buses are 0.95 p.u. and the upper limits are 1.1 for all the PV buses and 1.05 p.u. for all the PQ buses and the reference bus. The simulation results have been presented in Tables 1, 2, 3 &4. And in the Table 5 shows the proposed algorithm powerfully reduces the real power losses when compared to other given algorithms. The optimal values of the control variables along with the minimum loss obtained are given in Table 1. Corresponding to this control variable setting, it was found that there are no limit violations in any of the state variables.

<table>
<thead>
<tr>
<th>Control Variables</th>
<th>Variable Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.047</td>
</tr>
<tr>
<td>V2</td>
<td>1.044</td>
</tr>
<tr>
<td>V5</td>
<td>1.046</td>
</tr>
<tr>
<td>V8</td>
<td>1.033</td>
</tr>
<tr>
<td>V11</td>
<td>1.002</td>
</tr>
</tbody>
</table>
Optimal Reactive Power Dispatch problem together with voltage stability constraint problem was handled in this case as a multi-objective optimization problem where both power loss and maximum voltage stability margin of the system were optimized simultaneously. Table 2 indicates the optimal values of these control variables. Also it is found that there are no limit violations of the state variables. It indicates the voltage stability index has increased from 0.2486 to 0.2498, an advance in the system voltage stability. To determine the voltage security of the system, contingency analysis was conducted using the control variable setting obtained in case 1 and case 2. The Eigen values equivalents to the four critical contingencies are given in Table 3. From this result it is observed that the Eigen value has been improved considerably for all contingencies in the second case.

Table 2: Results of COS - Voltage Stability Control Reactive Power Dispatch Optimal Control Variables

<table>
<thead>
<tr>
<th>Control Variables</th>
<th>Variable Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.048</td>
</tr>
<tr>
<td>V2</td>
<td>1.047</td>
</tr>
<tr>
<td>V5</td>
<td>1.049</td>
</tr>
<tr>
<td>V8</td>
<td>1.034</td>
</tr>
<tr>
<td>V11</td>
<td>1.005</td>
</tr>
<tr>
<td>V13</td>
<td>1.038</td>
</tr>
<tr>
<td>T11</td>
<td>0.090</td>
</tr>
<tr>
<td>T12</td>
<td>0.090</td>
</tr>
<tr>
<td>T15</td>
<td>0.090</td>
</tr>
<tr>
<td>T36</td>
<td>0.090</td>
</tr>
<tr>
<td>Qc10</td>
<td>3</td>
</tr>
<tr>
<td>Qc12</td>
<td>2</td>
</tr>
<tr>
<td>Qc15</td>
<td>2</td>
</tr>
<tr>
<td>Qc17</td>
<td>3</td>
</tr>
<tr>
<td>Qc20</td>
<td>0</td>
</tr>
<tr>
<td>Qc23</td>
<td>2</td>
</tr>
</tbody>
</table>
Qc24
Qc29
Real power loss
SVSM

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Contingency</th>
<th>ORPD Setting</th>
<th>VSCRPD Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28-27</td>
<td>0.1419</td>
<td>0.1434</td>
</tr>
<tr>
<td>2</td>
<td>4-12</td>
<td>0.1642</td>
<td>0.1650</td>
</tr>
<tr>
<td>3</td>
<td>1-3</td>
<td>0.1761</td>
<td>0.1772</td>
</tr>
<tr>
<td>4</td>
<td>2-4</td>
<td>0.2022</td>
<td>0.2043</td>
</tr>
</tbody>
</table>

Table 3: Voltage Stability under Contingency State

<table>
<thead>
<tr>
<th>State variables</th>
<th>Limits</th>
<th>ORPD</th>
<th>VSCRPD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower</td>
<td>Upper</td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>-20</td>
<td>152</td>
<td>1.3422</td>
</tr>
<tr>
<td>Q2</td>
<td>-20</td>
<td>61</td>
<td>8.9900</td>
</tr>
<tr>
<td>Q5</td>
<td>-15</td>
<td>49.92</td>
<td>25.920</td>
</tr>
<tr>
<td>Q8</td>
<td>-10</td>
<td>63.52</td>
<td>38.820</td>
</tr>
<tr>
<td>Q11</td>
<td>-15</td>
<td>42</td>
<td>2.9300</td>
</tr>
<tr>
<td>Q13</td>
<td>-15</td>
<td>48</td>
<td>8.1025</td>
</tr>
<tr>
<td>V3</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0372</td>
</tr>
<tr>
<td>V4</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0307</td>
</tr>
<tr>
<td>V6</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0282</td>
</tr>
<tr>
<td>V7</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0101</td>
</tr>
<tr>
<td>V9</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0462</td>
</tr>
<tr>
<td>V10</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0482</td>
</tr>
<tr>
<td>V12</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0400</td>
</tr>
<tr>
<td>V14</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0474</td>
</tr>
<tr>
<td>V15</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0457</td>
</tr>
<tr>
<td>V16</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0426</td>
</tr>
<tr>
<td>V17</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0382</td>
</tr>
<tr>
<td>V18</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0392</td>
</tr>
<tr>
<td>V19</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0381</td>
</tr>
<tr>
<td>V20</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0112</td>
</tr>
<tr>
<td>V21</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0435</td>
</tr>
<tr>
<td>V22</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0448</td>
</tr>
<tr>
<td>V23</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0472</td>
</tr>
<tr>
<td>V24</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0484</td>
</tr>
<tr>
<td>V25</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0142</td>
</tr>
<tr>
<td>V26</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0494</td>
</tr>
<tr>
<td>V27</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0472</td>
</tr>
<tr>
<td>V28</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0243</td>
</tr>
<tr>
<td>V29</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0439</td>
</tr>
<tr>
<td>V30</td>
<td>0.95</td>
<td>1.05</td>
<td>1.0418</td>
</tr>
</tbody>
</table>
Table 5: Comparison of Real Power Loss

<table>
<thead>
<tr>
<th>Method</th>
<th>Minimum loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolutionary programming [26]</td>
<td>5.0159</td>
</tr>
<tr>
<td>Genetic algorithm [27]</td>
<td>4.665</td>
</tr>
<tr>
<td>Real coded GA with Lindex as SVSM [28]</td>
<td>4.568</td>
</tr>
<tr>
<td>Real coded genetic algorithm [29]</td>
<td>4.5015</td>
</tr>
<tr>
<td>Proposed COS method</td>
<td>4.2796</td>
</tr>
</tbody>
</table>

7. Conclusion

In this paper, the Condition of Substance Search (COS) algorithm has been successfully implemented to solve Optimal Reactive Power Dispatch problem. The proposed algorithm has been tested in standard IEEE 30-bus system. The results are compared with other heuristic methods and the proposed Condition of Substance Search (COS) algorithm demonstrated its effectiveness in minimization of real power loss & voltage stability also enhanced. Simulation Results clearly shows various system control variables are well within the acceptable limits.

References


*Corresponding author.
E-mail address: gklenin@ gmail.com