Abstract—Many heuristic optimization techniques derived from evolutionary algorithms are stochastic search methods that mimic the natural biological evolution and/or social behavior of spices. Such algorithms have been developed to obtain near optimal solutions to large scale optimization problems, where in traditional methods may fail. With the development of computational intelligence in recent years, the area of artificial immune systems (AIS) greatly influencing the engineering applications. This paper presents the development and comparative application of recently developed artificial immune system (AIS) based Clonal section algorithm and its variant (adaptive Clonal selection algorithm) for solving single objective optimal power flow (OPF) problems. This problem is formulated as optimization of cost, loss and L-index objectives individually by considering various security constraints. In order to study the effectiveness of the proposed methods, they are tested on standard IEEE 30-bus test system. Based on the comparative results, it is found adaptive Clonal selection algorithm (ACSA) performs better than the basic CSA.

I. INTRODUCTION

For modern power system operation, control and planning one of the most significant studies required is an optimal power flow. Optimal power flow could be treated as a static nonlinear programming problem. The main objective of the OPF problem is to determine the optimal operating state of the power system by optimizing a nonlinear objective function while satisfying certain equality and inequality constraints. Many optimization techniques both conventional and non-conventional methods are applied to solve OPF as reported in literature. However, the conventional methods such as Newton’s method [1], gradient methods [2, 3], linear programming [4, 5], dynamic programming and interior point methods [6, 7] are often facing problem of convergence and difficulties in obtaining the global optimum. Since, these methods rely on initial point and convexity to obtain the global optimum solution.

The main drawbacks of linear programming based techniques are insecure convergence characteristics and complexity of algorithm. Similarly Quadratic programming based techniques have some disadvantages associated with the piecewise quadratic cost approximation. The convergence characteristics of Newton-based algorithm are more sensitive to initial conditions. And sometimes this method may also fail to converge due to improper initial conditions. Numerical difficulties may arise in sequential unconstrained optimization techniques due to large penalty factors. The interior point method is computationally efficient, but the main drawback is the solution may goes to infeasible region due to improper selection of step size [13].

Therefore, from the last few years many non-conventional methods have been developed and applied to solve the OPF problem by overcoming the limitations and drawbacks of classical optimization techniques. Recently, Geem et al. [8] proposed a new Harmonic Search (HS) meta-heuristic algorithm that was inspired by musical process of searching for a perfect state of harmony. HS has been successfully applied to many optimization problems [9-12]. It has been demonstrated in the literature that the search ability of evolutionary optimization algorithms can be improved by the hybridization with local search. Such a hybrid algorithm has been often referred to as memetic algorithms [14]. Memetic algorithms are hybrid algorithms that combine both the evolutionary algorithms and local search techniques.

In [15], K. M. Passino proposed an optimization technique known as bacterial foraging optimization algorithm based on the foraging strategies of the E. Coli bacterium cells. There have been a few successful applications of this algorithm in optimal control engineering, harmonic estimation, transmission loss reduction, machine learning etc. Shuffled frog leaping algorithm (SFLA) [16], is a newly developed memetic metaheuristic technique for combinatorial optimization, which has simple concept, few parameters, high performance, and easy programming. The SFLA has been tested on several benchmark functions that present its efficiency to many global optimization problems [17, 18]. Its effectiveness and suitability have also been demonstrated when applied to bridge deck repair problem [19]. Clonal Selection Algorithms (CSAs) are a special class of Immune algorithms (IA) which are inspired by the Clonal Selection Principle [20-22] of the human immune system to produce effective methods for search and optimization.

In this paper, Clonal selection algorithm is extended to solve single objective optimization problem with an adaptive feature. The proposed ACSA has been tested on a standard IEEE 30-bus test system for single objective with various case
II. OPTIMAL POWER FLOW PROBLEM FORMULATION

The main goal of the OPF problem is to optimize a selected objective function via optimal adjustment of the power system control variables, subject to several equality and inequality constraints. The optimal power flow problem can be mathematically formulated as follows:

Minimize \( F(x,u) \) 
Subject to:

1) Equality constraints:
   \[ g(x,u) = 0 \]  
   \[ h_{\text{min}} \leq h(x,u) \leq h_{\text{max}} \]  

where:
- \( F \) - objective function to be minimized
- \( x \) - vector of dependent variables (state variables)
- \( u \) - vector representing all control variables
- \( g \) - equality constraints representing system operating limits
- \( h \) - inequality constraints representing line flow equations

A. Objective function

1) Minimization of generation cost: The aim of this objective is to minimize the total cost \( F_g \) which is modeled as a quadratic cost curve function and can be represented as
   \[ F_g = \sum_{i=1}^{NG} a_i P_{gi}^2 + b_i P_{gi} + c_i \]  

where:
- \( a_i, b_i \), and \( c_i \) are the fuel cost coefficients of the \( i^{th} \) generator;
- \( P_{gi} \) is the real power output of the \( i^{th} \) generator.

2) Minimization of real power loss: This objective is to minimize the real power transmission line losses \( P_L \) in the system which can be expressed as follows:
   \[ P_L = \sum_{k=1}^{NL} g_k[V_i^2 + V_j^2 - 2V_i V_j \cos(\delta - \delta_j)] \]  

where:
- \( g_k \) is the conductance of a transmission line \( k \);
- \( V_i, V_j \) are the voltage magnitudes at bus \( i \) & \( j \) respectively;
- \( \delta_i, \delta_j \) are the phase angles at bus \( i \) & \( j \) respectively.

3) Minimization of voltage stability index (L-index): This objective is to maintain the voltage stability and move the system far away from the voltage collapse point. This can be achieved by minimizing the voltage stability indicator \( L \)-index [23] and can be expressed as
   \[ L_{\text{max}} = \max\{L_k : k = 1,2,...,N_{RQ} \} \]  

4) State Variables: The state variables of a power system network \( x \) consists of active power at slack bus \( (P_{g_i}) \), Voltage angle of load buses \( (V, \delta) \), Generator reactive power outputs \( (Q_{gi}) \), Transmission line loadings \( (S_{li}) \).

Hence, the state vector \( x \) can be represented as:
   \[ x = [P_{g1}, V_1, ..., V_{NRQ}, Q_{g1}, ..., Q_{NG}, S_{L1}, ..., S_{NL}] \]  

where:
- \( N_{RQ}, NG \) and \( NL \) are representing the number of load buses, generators and transmission lines respectively.

5) Control Variables: The control variables of power system network \( u \) consists of active power generation output \( (P_{gi}) \), generator voltage magnitude \( (V_{gi}) \), Tap settings of Transformer \( (T) \), and Shunt reactive power injections \( (Q_{gi}) \).

Hence, the vector \( U \) can be represented as:
   \[ U^T = [P_{g2}, ..., P_{gNG}, V_{g1}, ..., V_{gNG}, T_1, ..., T_{NT}, Q_{g1}, ..., Q_{gNC}] \]  

where:
- \( NT \) and \( NC \) are number of tap changing transformers and shunt reactive power injections respectively.

B. Constraints

1) Equality constraints: These constraints are described as typical load flow equations as follows:
   \[ P_{gi} - P_{gi} - \sum_{j=1}^{NB} V_j(G_{ji} \cos \delta_j + B_{ji} \sin \delta_j) = 0 \quad i \in NB \]  
   \[ Q_{gi} - Q_{gi} - \sum_{j=1}^{NB} V_j(G_{ji} \sin \delta_j - B_{ji} \cos \delta_j) = 0 \quad i \in NB \]  

where:
- \( P_{gi} \), & \( Q_{gi} \), are real and reactive power generations at \( i^{th} \) bus;
- \( P_{gi} \), & \( Q_{gi} \), are the real and reactive power demands at \( i^{th} \) bus;
- \( G_{ji} \), & \( B_{ji} \), are the conductance & susceptance of line \( i - j \);
- \( NB \) is the total number of buses.

2) Inequality constraints: These constraints represent the system operating limits as follows

Generation limits: Generator voltages, real power outputs and reactive power outputs are restricted by their lower and upper bounds as follows:
   \[ V_{gi}^{\text{min}} \leq V_{gi} \leq V_{gi}^{\text{max}}, \quad i = 1,...,NG \]  
   \[ P_{gi}^{\text{min}} \leq P_{gi} \leq P_{gi}^{\text{max}}, \quad i = 1,...,NG \]  
   \[ Q_{gi}^{\text{min}} \leq Q_{gi} \leq Q_{gi}^{\text{max}}, \quad i = 1,...,NG \]
Transformer constraints: Transformer tap settings are restricted by their minimum and maximum limits as follows:

\[ T_{i}^{\text{min}} \leq T_{i} \leq T_{i}^{\text{max}}, \quad i = 1, \ldots, NT \]  

(14)

Shunt VAR constraints: Reactive power injections at buses are restricted by their minimum and maximum limits as:

\[ Q_{i}^{\text{min}} \leq Q_{i} \leq Q_{i}^{\text{max}}, \quad i = 1, \ldots, NC \]  

(15)

Security constraints: These include the constraints of voltage magnitudes at load buses and transmission line loadings as follows:

\[ V_{i}^{\text{min}} \leq V_{i} \leq V_{i}^{\text{max}}, \quad i = 1, \ldots, N_{L} \]  

\[ S_{i} \leq S_{i}^{\text{max}}, \quad i = 1, \ldots, NL \]  

(17)

3) Constraints handling: There are different strategies to handle constraints in evolutionary computation optimization algorithms. In this paper, the constraints are incorporated into fitness function by means of a penalty function method. In this method a penalty factor multiplied with the violated value of variable is added to the objective function and any infeasible solution obtained is rejected.

III. OVERVIEW OF IMMUNE ALGORITHMS

A. Clonal selection algorithm

Clonal selection theory is one of the fundamental models used to explain the behaviours of the modern immune system. The biological principles like clone generation, proliferation and maturation are mimicked and incorporated into an artificial immune based algorithm termed the Clonal selection algorithm [20, 21]. The Clonal selection algorithm (CSA) named CLONALG, was proposed by Leandro N de Castro and Fernando J Von Zuben [21], is a population based stochastic method. This CSA is more extensively used artificial immune based optimization method in pattern recognition and multimodal optimization problems with binary representation of variables. The implementation of Clonal selection algorithm for artificial immune system involves the following four stages [20].

- initialization of antibodies
- cloning and selection (proliferation and differentiation on the encounter of cells with antigens)
- maturation and diversification of antibody types by carrying out affinity maturation process through random genetic changes
- removal of differentiated immune cells which posses low affinity.

In the process of optimization with AIS based Clonal selection algorithm, affinity is evaluated using fitness or objective function value while satisfying constraints. Here the constraints are represented by antigens and the constraint satisfaction attained by antibody – antigen affinity. In other words more the affinity higher is the constraint satisfaction. Apart from the above, if two solutions equally satisfy their constraints, one with better value of the corresponding objective attains larger affinity or fitness value.

Step-by-step procedure:

Step 1: Initially generate a set \( (N) \) of candidate solutions, which consists of the subset of memory cells \( (M) \) added to the remaining \( (N_{i}) \) population \( (N = N_{i} + M) \);

Step 2: Find out (Select) the \( N_{i} \) best individuals of the population \( (P_{i}) \), based on their affinity value;

Step 3: Proliferate (Clone) these \( N_{i} \) best individuals of the population, results in temporary population of clones \( (N_{c}) \). The clone size is an increasing function with their affinity

Step 4: Now send the population of clones to a hypermutation process, where the hypermutation rate is proportional to their affinity and a maturated antibody population is generated \( (N_{c}') \)

Step 5: Re-select the improved individuals from \( N_{c}' \) to compose the new memory set \( M \). Some members of \( N \) can be replaced with these improved members of \( N_{c}' \);

Step 6: Replace \( N_{j} \) antibodies by diversity introduction with random generation. The cells which are having lower affinity have higher probabilities of being replaced.

B. Adaptive Clonal selection algorithm

In Clonal selection algorithm the number of clones generated per antibody is dependent on the affinity or fitness value. The cloning population size is evaluated using Eq. (18)

\[ NC = \sum_{i=1}^{N_{i}} NC_{i} \]  

(18)

Each term in summation of above equation represents clone size of selected antibody ‘\( i \)’ and is given by

\[ NC_{i} = \text{round} \left( \frac{\beta N_{i} cc}{i} \right) \]  

(19)
Where
\[ \beta \] is the multiplication factor of clone size;
\[ N_{op} \] - Population size and
\[ \gamma \] - is accelerating factor.

The accelerating factor "cc" has been considered as fixed value 1, for all generations in conventional CSA. While in adaptive Clonal selection algorithm "cc" value is updated dynamically so that its value is varying for every generation. At the end of every generation the acceleration factor is updated as \[ cc = cc \times \gamma \]. The initial value of \( cc < 1 \) and \( \gamma \) value lies between 0.5 and 1.1.

The idea here is to use dynamic accelerating factor instead of fixed value to obtain best Clonal population size \( N_c \). With the increase in generation count the Cloning population size decreases so that fast convergence can be obtained. The implementation steps of this adaptive feature are shown in the algorithm described in the following section.

IV. IMPLEMENTATION OF CLONAL SELECTION ALGORITHM FOR OPTIMAL POWER FLOW PROBLEM

A. Initialization and Cloning

Initially a population of antibodies is generated randomly using real coded numbers. That is, the generator real power output, generator voltages, tap changer positions and reactive power injections at specified buses is formed as one antibody. Each antibody is checked for constraint violation and penalized in case of infeasibility with penalty proportional to the extent of constraint violation. Each of the antibodies from the initial pool is copied into a fixed number of clones to generate a temporary population of clones. This population of clones is made to undergo maturation process by means of hypermutation mechanism.

The hypermutation is carried out using affinity based hypermutation rate. That means larger hypermutation rate is set for lower affinity clones and smaller hypermutation for high affinity clones, i.e. the probability of hypermutation of each clone is inversely proportional to its affinity. Then evaluation of affinity corresponding to mutated clones and apply penalty in case of any constraint violation. Now, the new population of antibodies with same size of initial population is selected from the mutated clones and this completes the one iteration. In the next iteration, this fresh population is made to undergo cloning and hyper mutation as discussed above and likewise.

B. Affinity Evaluation and Maturation

The number of molecules in each antibody i.e. solution to the OPF problem is chosen as \( N \). It indicates the total number of control variables of the problem. The size of antibody population is assumed to be 40 and the number of clones generated per antibody is kept dependent on its fitness value. The hyper-mutation is performed through Rand/2 hypermutation strategy as described in [24]. The probability of hypermutation is also depends on the fitness of a clone. Affinity evaluation is done by converting the cost function into fitness function Eq. (20), thereby calculating the fitness value required to carry out a proportional selection of mutated clones. The fitness function to evaluate the affinity of any antibody is given by the following expression

\[ F = \frac{1}{J + J_{pen}} \]  \hspace{1cm} (20)

where
- \( F \) is the fitness function of an antibody \( X \);
- \( J \) is the cost associated with that antibody and
- \( J_{pen} \) is the sum of all penalties for security constraint violations.

The number of clones generated per antibody is dependent in our solution methodology on the affinity or fitness value, i.e., larger number of clones is generated for the antibodies with higher fitness value.

The mutation rate is not taken uniform but kept inversely proportional to the fitness value of a given clone. Consequently, clones with higher fitness are made liable to undergo mutation to a lesser extent as compared to those with lower fitness. Thereafter evaluate the affinity of mutated clones. This is repeated till all the clones from the temporary Clonal population are endured to mutation.

Finally, tournament selection is done to select same number of mutated clones as there are in initial population. This completes one generation of the Clonal selection algorithm. The convergence parameter is set as the situation when the best solutions of each generation cease to change. Thereby, stopping criteria is taken as convergence or the number of cycles (i.e., the one which is achieved first) subjected to a maximum of 500 cycles.

V. SIMULATION RESULTS AND DISCUSSIONS

A. IEEE 30-Bus System Results

To test the effectiveness of the proposed CSA and ACSA methods for an optimal power flow solution a modified IEEE 30-bus test system is considered. The test system consists of six generators at buses 1, 2, 3, 4, 5 and 6 and four transformers with off-nominal tap ratio in the lines 13–9, 13–10, 11–12 & 27–28 and nine shunt reactive power injection at the buses 10, 12, 15, 17, 20, 21, 23, 24 and 29. The complete system data is taken from reference [25]. All the generator voltages and active powers (except slack bus 1), and tap settings of the regulating transformers and VAR injections are considered as continuous variables.

The minimum and maximum limits of the control variables are given in Table I.

<table>
<thead>
<tr>
<th>Variables</th>
<th>( V_{gs} )</th>
<th>Taps</th>
<th>( Q_{shunt} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>0.95</td>
<td>0.9</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>1.1</td>
<td>1.1</td>
<td>0.05</td>
</tr>
</tbody>
</table>

In order to demonstrate the effectiveness and robustness of the proposed techniques the following three cases with different objectives are simulated on the test system.
Case 1: Minimization of fuel cost: In this case, the proposed CSA and ACSA algorithms are applied for solving the optimal power flow problem with different equality and inequality constraints by considering cost function described in section III. Initially, before optimization, the total fuel cost of the test system was 901.3188 $/h with base case control variables. The base case load flow solution result is shown in Table II. The convergence characteristics of fuel cost for both the methods are shown in Fig 2(a). It is clearly indicating that the convergence of ACSA method is faster than CSA even though the optimal cost in both the methods is approximately same. The optimal settings of the control variables with the proposed methods for cost objective are shown in Table II. The total fuel cost obtained by the proposed CSA and ACSA methods are 799.0796$/h and 799.0351$/h respectively. It can be noticed that the reduction in fuel cost by 11.34% while the system loss is increased by 52.88% of its base case value and L-index decrease by 21.3% of its base case value.

Case 2: Minimization of loss: In this case the objective function $L_P$ described in Eq. (5) is optimized with the proposed methods. All the constraints described in problem formulation are considered. The convergence characteristics for loss minimization with the proposed methods are shown in Fig 2(b). The optimal settings of the control variables corresponding to case 2 are also given in Table II. The total loss obtained by the proposed CSA and ACSA methods are 2.8699MW and 2.8627MW respectively. The minimization of the loss objective results in the reduction of loss by 48.68% whereas the fuel cost increased by 6.88% when compared to base case value.

Case 3: Minimization of L-index: Voltage stability enhancement is another important aspect with reference to power system operation. Hence the minimization is L-index is considered as objective function stated in Eq. (6). The convergence characteristics for L-index optimization obtained with the proposed methods are shown in Fig 2(c). The optimal settings of the control variables corresponding to case 3 are also given in Table II for both the methods. From the results it is noticed that there is significant reduction in L-index value as 0.1095p.u with ACSA method when compared to base case value.

In each case the minimum and maximum values of bus voltages are also tabulated in Table II, they are maintained within the limits. Computation time required per iteration is comparatively less in case of ACSA method. The results obtained from the proposed methods are also compared with various methods reported in the literature are summarized in Table III. Some of these methods are Differential Evolution (DE) [26], Particle swarm optimization (PSO) [27], enhanced genetic algorithm (EGADQLF) [27], PSO [28] and IPSO [28] for the cost, loss and L-index objective functions. It is clear that the proposed method can achieve better results when compared to other algorithms reported in the literature for cost objective and loss objective. In case of L-index optimization the result obtained with ACSA method is better than all other methods reported in literature except EGADQLF [27].

![Fig. 2(a-c): Convergence characteristics for single objectives](image_url)
TABLE II
OPTIMAL SOLUTION FOR SINGLE OBJECTIVES WITH CSA AND ITS VARIANT ACSA METHODS

<table>
<thead>
<tr>
<th>Control Variables</th>
<th>Base Case</th>
<th>Case-1</th>
<th>Case-2</th>
<th>Case-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CSA</td>
<td>ACSA</td>
<td>CSA</td>
</tr>
<tr>
<td>P_{g1}(MW)</td>
<td>98.9924</td>
<td>175.29</td>
<td>177.07</td>
<td>52.8699</td>
</tr>
<tr>
<td>P_{g2}(MW)</td>
<td>80.0000</td>
<td>48.96</td>
<td>48.69</td>
<td>78.4542</td>
</tr>
<tr>
<td>P_{g3}(MW)</td>
<td>20.0000</td>
<td>22.29</td>
<td>21.05</td>
<td>34.9646</td>
</tr>
<tr>
<td>P_{g4}(MW)</td>
<td>20.0000</td>
<td>12.07</td>
<td>11.90</td>
<td>30.0000</td>
</tr>
<tr>
<td>P_{g5}(MW)</td>
<td>50.0000</td>
<td>21.27</td>
<td>21.30</td>
<td>50.0000</td>
</tr>
<tr>
<td>P_{g6}(MW)</td>
<td>20.0000</td>
<td>12.02</td>
<td>12.00</td>
<td>39.9994</td>
</tr>
<tr>
<td>V_{1}(p.u.)</td>
<td>1.0500</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0099</td>
</tr>
<tr>
<td>V_{2}(p.u.)</td>
<td>1.0450</td>
<td>1.0874</td>
<td>1.0878</td>
<td>1.0971</td>
</tr>
<tr>
<td>V_{3}(p.u.)</td>
<td>1.0100</td>
<td>1.0698</td>
<td>1.0691</td>
<td>1.0872</td>
</tr>
<tr>
<td>V_{4}(p.u.)</td>
<td>1.0500</td>
<td>1.0833</td>
<td>1.0849</td>
<td>1.0658</td>
</tr>
<tr>
<td>V_{5}(p.u.)</td>
<td>1.0100</td>
<td>1.0621</td>
<td>1.0618</td>
<td>1.0801</td>
</tr>
<tr>
<td>V_{6}(p.u.)</td>
<td>1.0500</td>
<td>1.0999</td>
<td>1.1000</td>
<td>1.1000</td>
</tr>
<tr>
<td>T_{1}</td>
<td>0.9780</td>
<td>0.9966</td>
<td>0.9694</td>
<td>0.9370</td>
</tr>
<tr>
<td>T_{2}</td>
<td>0.9690</td>
<td>1.0176</td>
<td>0.9062</td>
<td>0.9371</td>
</tr>
<tr>
<td>T_{3}</td>
<td>0.9320</td>
<td>0.9798</td>
<td>0.9679</td>
<td>0.9829</td>
</tr>
<tr>
<td>T_{4}</td>
<td>0.9680</td>
<td>0.9682</td>
<td>0.9594</td>
<td>0.9799</td>
</tr>
<tr>
<td>Q_{1,1}(p.u.)</td>
<td>0.0294</td>
<td>0.0006</td>
<td>0.0242</td>
<td>0.0042</td>
</tr>
<tr>
<td>Q_{1,2}(p.u.)</td>
<td>0.0485</td>
<td>0.0500</td>
<td>0.0488</td>
<td>0.0500</td>
</tr>
<tr>
<td>Q_{1,5}(p.u.)</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
<tr>
<td>Q_{1,7}(p.u.)</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0469</td>
<td>0.0500</td>
</tr>
<tr>
<td>Q_{1,8}(p.u.)</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
<td>0.0500</td>
</tr>
<tr>
<td>Q_{2,2}(p.u.)</td>
<td>0.0467</td>
<td>0.0493</td>
<td>0.0402</td>
<td>0.0314</td>
</tr>
<tr>
<td>Q_{2,3}(p.u.)</td>
<td>0.0308</td>
<td>0.0247</td>
<td>0.0118</td>
<td>0.0192</td>
</tr>
<tr>
<td>Q_{2,4}(p.u.)</td>
<td>0.0219</td>
<td>0.0269</td>
<td>0.0397</td>
<td>0.0236</td>
</tr>
<tr>
<td>Q_{2,9}(p.u.)</td>
<td>0.0248</td>
<td>0.0243</td>
<td>0.0170</td>
<td>0.0182</td>
</tr>
<tr>
<td>Fuel Cost ($/h)</td>
<td>901.3188</td>
<td>799.0796</td>
<td>799.0351</td>
<td>963.7385</td>
</tr>
<tr>
<td>Loss (MW)</td>
<td>5.5924</td>
<td>8.50</td>
<td>8.62</td>
<td>2.86999</td>
</tr>
<tr>
<td>L-index</td>
<td>0.1489</td>
<td>0.1176</td>
<td>0.1165</td>
<td>0.11800</td>
</tr>
<tr>
<td>V_{min}</td>
<td>0.9734</td>
<td>1.0600</td>
<td>1.0596</td>
<td>1.06343</td>
</tr>
<tr>
<td>V_{max}</td>
<td>1.0500</td>
<td>1.1000</td>
<td>1.1000</td>
<td>1.09955</td>
</tr>
<tr>
<td>No of iter.</td>
<td>3</td>
<td>450</td>
<td>100</td>
<td>435</td>
</tr>
<tr>
<td>time/iter.(s)</td>
<td>0.0478</td>
<td>1.71092</td>
<td>0.760152</td>
<td>2.09250</td>
</tr>
</tbody>
</table>

TABLE III
COMPARISON OF RESULTS WITH DIFFERENT METHODS REPORTED IN LITERATURE

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost ($/h)</th>
<th>Loss (MW)</th>
<th>L-index</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE [26]</td>
<td>800.56</td>
<td>3.24</td>
<td>0.1246</td>
</tr>
<tr>
<td>PSO [27]</td>
<td>802.19</td>
<td>3.6294</td>
<td>0.11055</td>
</tr>
<tr>
<td>EGADQLF [27]</td>
<td>799.56</td>
<td>3.2088</td>
<td>0.10402</td>
</tr>
<tr>
<td>PSO [28]</td>
<td>802.205</td>
<td>5.2115</td>
<td>0.1307</td>
</tr>
<tr>
<td>IPSO [28]</td>
<td>801.978</td>
<td>5.0732</td>
<td>0.1037</td>
</tr>
<tr>
<td>CSA (proposed)</td>
<td>799.0796</td>
<td>2.86999</td>
<td>0.1134</td>
</tr>
<tr>
<td>ACSA (proposed)</td>
<td>799.0351</td>
<td>2.8627</td>
<td>0.1095</td>
</tr>
</tbody>
</table>

B. Effect of algorithm parameter variation

The effect of variation of population size and adaptive factors of the proposed ACSA method on solution quality is also studied for cost objective. Comparison of convergence characteristics for variation of adaptive factor and population size are shown in Fig 3 and Fig 4 respectively. From Fig 3, it is observed that the effect of variation of adaptive factor causes trivial change in final solution. However the population size does not have significant change in optimal solution as seen in the Fig 4. The optimal solution results for parameter variation are shown in Table IV. High value of adaptive factor results in best value of optimal solution as summarised in Table IV.

Fig. 3: Cost Convergence characteristics with different adaptive factors
VI. CONCLUSIONS

In this paper, Clonal selection and adaptive Clonal selection algorithms have been developed and successfully applied to solve single objective optimal power flow problem with different objectives. The implementation steps for each method for OPF solution are also presented. MATLAB programs were written to implement each algorithm. Comparison application of these algorithms with different objectives on IEEE 30-bus test system is presented. From the result analysis it was found that Clonal selection algorithm with adaptive feature appears to be work good for optimal power flow problem with various objectives like fuel cost, loss and L-index. Finally the results obtained with the proposed ACSA method were also compared with those reported in the literature for all the 3-cases. From the comparison of results, the performance of ACSA method was found better than other methods in terms of solution quality, and computational time for optimal power flow problem.

REFERENCE