

COMPARISON OF OPTIMIZATION TECHNIQUE TO FIND THE OPTIMAL LOCATION OF FACTS CONTROLLERS FOR TRANSMISSION LINE

¹Jagadeesh Kumar Muthukrishnan, ¹Subhransu Sekhar Dash,
¹Harish Kiran Selvakumar, ¹Subramani Chinnamuthu and ²Palanivel Panjamoorthy

¹Department of EEE, SRM University Chennai, Tamil Nadu, India

²Department of EEE, M.A.M College of Engineering, Tiruchirappalli, Tamil Nadu, India

Received 2013-07-08; Revised 2013-07-27; Accepted 2013-12-26

ABSTRACT

The aspect of this study is to compare the three different algorithms such as Genetic Algorithm (GA), Enhanced Genetic Algorithm (EGA) and Particle Swarm Optimization (PSO) to find the optimal location of the FACTS device in the transmission line. The FACTS devices Thyristor Controlled Series Compensator (TCSC) and Unified Power Flow Controller (UPFC) are selected based on the steady state stability and dynamic stability state. Each of the algorithms is coded in MATLAB and incorporated it with the conventional Newton Raphson's load flow analysis. To find the effectiveness of the proposed algorithm IEEE 14-bus system is taken as the test system.

Keywords: Optimal Power Flow (OPF), Flexible AC Transmission System (FACTS), Genetic Algorithm (GA), Enhanced Genetic Algorithm (EGA), Particle Swarm Optimization (PSO), Newton Raphson's (NR) Power Flow

1. INTRODUCTION

The power flow in the power system has become very complex. In order to have a maximum power transfer and utilization of the existing power system resources. This can be achieved by the use of the FACTS devices in the power system.

The various improvements when the FACTS devices are connected in the power system network are the stability, behaviour and the reliability of the power system. The dynamic and the steady state stability possible problems and the devices to be used to overcome are discussed. For the selection of the FACTS devices the other criteria necessary are its voltage limits, thermal limits, loop flow, short circuit level and the sub-synchronous resonance.

There are various researches in progress to improve the static performance of the power system. The cost is a

main factor to select the FACTS devices. Hence the generation cost and the cost of device to be used. There are many works still in progress to find the location and the optimal choice of the device.

The main idea is to develop a MATLAB based code for the algorithms to find device and the rating to be connected along with it the generators limits such that the overall system cost including the device cost is minimized.

2. FACTS DEVICES SELECTION

The various possible problems that can occur in the power system are tabulated in **Table 1 and 2** as steady state stability and dynamic stability state (Kiran *et al.*, 2011). The FACTS devices can be classified based on who they are connected to the power system as shunt, series or both together. The different types of FACTS devices are TCSC, TSSC, UPFC, SVC, STATCOM and some other devices.

Corresponding Author: Jagadeesh Kumar Muthukrishnan, Department of EEE, SRM University Chennai, Tamil Nadu, India

Table 1. Dynamic stability state

Issue	Type of system	Corrective action	FACTS controller
Transient stability	Radial, Interconnected, loosely meshed network	Increase synchronizing torque	TCSC, TSSC, UPFC
	Interconnected, loosely and tightly meshed network	Dynamic load flow	TCPAR, UPFC, TCSC
Dampening	Radial, Interconnected, loosely meshed network	Damper 1Hz oscillation	SVC, TCSC, STATCOM
		Damper low frequency oscillations	SVC, TCSC, UPFC, STATCOM
Post Contingency Voltage Control	Radial, Interconnected, loosely meshed network	Dynamic voltage support and flow control	SVC, UPFC, TCSC
	Radial, Interconnected, loosely and tightly meshed network	Reduced impact of contingency	SVC, TCSC, STATCOM, UPFC
Voltage Stability	Interconnected, loosely and tightly meshed network	Reactive support	UPFC, SVC, STATCOM
		Network control actions	UPFC, TCSC, STATCOM

Table 2. Steady state stability

Issue	Problem	Corrective action	FACTS controller used
Voltage limits	Low voltage at heavy load	Supply reactive power	SVC, TCSC, STATCOM
	High voltage at light load	Remove reactive power	TCSC, STATCOM
	Low voltage and outage	Prevent overload	TCPAR, TCSC
Thermal limits over load	Line or transformer	Supply reactive power and limit over load	TCSC, UPFC, STATCOM, SVC
	Tripping of parallel circuit	Reduce overload	TCSC, UPFC, TCPAR
Loop flow	Parallel line load sharing	Limiting circuit loading	UPFC, TCSC
	Post-fault sharing	Adjust series reactance	UPFC, TCSC
	Flow direction reversal	Rearrange network	TCSC, UPFC, TCPAR
Short circuit level SSR	Excessive breaker fault current	Adjust phase angle	TCPAR, UPFC
	Generator shaft damage	Limit short circuit current	TCSC, UPFC
		Mitigate oscillations	TCSC

From the dynamic stability state tabulation it can be concluded that for most of the problems, the necessary corrective measure and the device to be connected to overcome the problem in most of the cases are:

- UPFC
- TCSC

From the tabulation in **Table 2** it can be concluded that for all the possible problems in steady state stability the same devices to be connected to overcome the problem in most of the cases:

- TCSC
- UPFC

In the analysis of the problems that are tabulated in the **Table 1 and 2** the best devices that can be used in the power system are UPFC and TCSC.

This study does not deal with modelling of the TCSC and UPFC. Only the device selection and the rating of the FACTS device to be used are obtained as the result.

3. COST FUNNCTION

As stated earlier the objective in this study is to reduce the overall cost function of both generation cost (Subramani *et al.*, 2012) and investment cost of the FACTS devices (Kiran *et al.*, 2012). Also, to find simultaneously the optimal choice and location of FACTS device so the overall cost is minimized.

3.1. Generation Cost Function

The costs function of the generation represented by a quadratic polynomial as follows Equation (1):

$$C_2(PG) = \alpha_0 + \alpha_1 PG + \alpha_2 PG^2 \tag{1}$$

whereas PG is the output of the generator (MW) and α_0 , α_1 and α_2 are cost coefficients.

3.2. Facts Controller Cost Function

Based the Siemens AG Database the cost function for the controller that has been selected to use are as follows:

The cost function for UPFC is Equation (2):

$$C_{1UPFC} = 0.0003s^2 - 0.2691s + 188.22 \text{ (US\$ / kvar)} \quad (2)$$

The cost function for TCSC is Equation (3):

$$C_{1TCSC} = 0.0015s^2 - 0.7130s + 153.75 \text{ (US\$ / kVar)} \quad (3)$$

The rating of the device is given by Equation (4 and 5):

$$R_{TCSC} = r_f * 0.45 - 0.25 \text{ (Mvar)} \quad (4)$$

$$R_{UPFC} = r_f * 180 \text{ (MVar)} \quad (5)$$

Where C_{1UPFC} and C_{1TCSC} are in US\$/kVar and s is the operating of the FACTS controller in MVar. The cost function of the TCSC and UPFC are found as shown in Fig. 1.

3.3. Optimal Power Flow with Facts Device

The total cost of the FACTS devices along with the generation cost is implemented as Equation (6 to 8):

$$\text{Minimum } C_{\text{Total}} = C_1(f) + C_2(PG) \quad (6)$$

$$\text{Subjected to } E(f, g) = 0 \quad (7)$$

$$B_1(f) < b_1, B_2(g) < b_2 \quad (8)$$

where, C_{Total} -the overall cost objective function which includes the average investment costs of FACTS devices $C_1(f)$ and the generation cost $C_2(PG)$.

$E(f, g)$ = The conventional power flow equations.

$B_1(f)$ and $B_2(g)$ = The inequality constraints for FACTS controllers and the conventional power flow respectively.

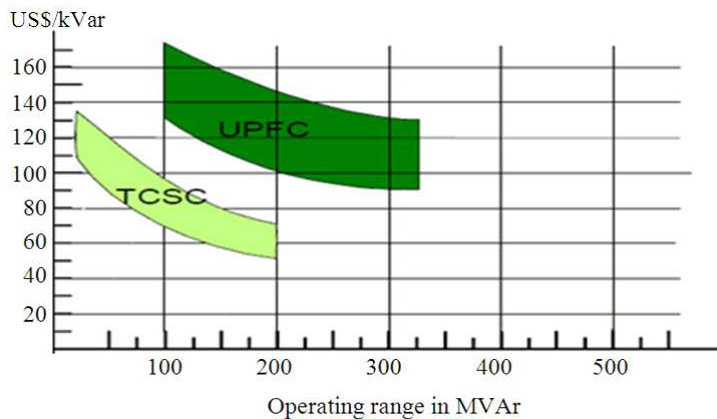
f & PG = vectors that represent the variables of FACTS controllers and the active power outputs of the generators.

g = Represents the operating state of the power system.

The unit for generation cost is US\$/Hour and the investment cost of FACTS controllers are US\$. They must be unified into US\$/Hour. Normally the FACTS controllers will be in service for many years. However only a part of its life time is employed to regulate the power flow. In this study three years is employed to evaluate the cost function. Therefore the average value of the investment costs are calculated as follows Equation (9):

$$C_1(f) = C(f) / \{8760 \times 3\} \quad (9)$$

As mentioned above, power system parameters can be changed using FACTS controllers. These different parameters derive different results on the objective function. Also, the variation of FACTS locations and FACTS types has also influences on the objective function. Therefore, using the conventional optimization methods is not easy to find the optimal location of FACTS devices, types and control parameters simultaneously. To solve this problem, optimization technique such as Genetic Algorithm, Enhanced Genetic Algorithm and Particle Swarm Algorithm are employed in conjunction with conventional NR power flow method.



Source: Siemens AG database

Fig. 1. Typical investment cost for TCSC and UPFC

4. GENETIC ALGORITHM

GA's are general purpose optimization algorithms (Popov and Hamburg, 2008) based on the mechanics of natural selection and genetics. They operate on string structures (chromosomes), typically a concatenated list of binary digits representing a coding of the control parameters (phenotype) of a given problem. Chromosomes themselves are composed of genes. The real value of a control parameter, encoded in a gene, is called an allele.

GA's are an attractive alternative to other optimization methods because of their robustness. There are three major differences between GAs and conventional optimization algorithms. First, GAs operates on the encoded string of the problem parameters rather than the actual parameters of the problem. Each string can be thought of as a chromosome that completely describes one candidate solution to the problem. Second, GAs uses a population of points rather than a single point in their search. This allows the GA to explore several areas of the search space simultaneously, reducing the probability of finding local optima. Third, GAs do not require any prior knowledge, space limitations, or special properties of the function to be optimized, such as smoothness, convexity or existence of derivatives. They only require the evaluation of the so-called Fitness Function (FF) to assign a quality value to every solution produced.

Assuming an initial random population produced and evaluated, genetic evolution takes place by means of three basic genetic operators:

- Parent selection
- Crossover
- Mutation

Parent selection is a simple procedure whereby two chromosomes are selected from the parent population based on their fitness value. Solutions with high fitness values have a high probability of contributing new offspring to the next generation. The selection rule used in our approach is a simple roulette-wheel selection.

Crossover is an extremely important operator for the GA. It is responsible for the structure recombination (information exchange between mating chromosomes) and the convergence speed of the GA and is usually applied with high probability (0.6-0.9). The chromosomes of the two parents selected are combined to form new chromosomes that inherit

segments of information stored in parent chromosomes. Until now, many crossover schemes, such as single point, multipoint, or uniform crossover have been proposed in the literature. Uniform crossover has been used in our implementation.

While crossover is the main genetic operator exploiting the information included in the current generation, it does not produce new information.

Mutation is the operator responsible for the injection of new information. With a small probability, random bits of the offspring chromosomes flip from 0 to 1 and vice versa and give new characteristics that do not exist in the parent population. In our approach, the mutation operator is applied with a relatively small probability (0.0001-0.001) to every bit of the chromosome.

The algorithm flow sequence of Simple genetic algorithm is as follows:

- Begin
- Initialization of generation and population
- Check for generation level
- Evaluation of the fitness function
- Select the best parent in the generation
- Crossover the parent generation to form a new generation
- Increases the generation by one move to step 3
- End

The FF evaluation and genetic evolution take part in an iterative procedure, which ends when a maximum number of generations are reached.

When applying GAs to solve a particular optimization problem (OPF in our case), two main issues must be addressed:

- The encoding, i.e., how the problem physical decision variables are translated to a GA chromosome and its inverse operator, decoding
- The definition of the FF to be maximized by the GA (the GA FF is formed by an appropriate transformation of the initial problem objective function augmented by penalty terms that penalize the violation of the problem constraints

4.1. Encoding

In the SGA, after the application of the basic genetic operators (parent selection, crossover and mutation) the advanced and problem-specific

operators are applied to produce the new generation. All chromosomes in the initial population are created at random (every bit in the chromosome has equal probability of being switched ON or OFF).

Due to the decoding process selection, the corresponding control variables of the initial population satisfy their upper–lower bound or discrete value constraints. Population statistics are then used to adaptively change the crossover and mutation probabilities. If premature convergence is detected the mutation probability is increased and the crossover probability is decreased. The contrary happens in the case of high population diversity.

4.2. Fitness Function

GAs is usually designed so as to maximize the FF, which is a measure of the quality of each candidate solution. The objective of the OPF problem is to minimize the total operating cost (6).

Therefore, a transformation is needed to convert the cost objective of the OPF problem to an appropriate FF to be maximized by the GA. The OPF functional operating constraints (9) are included in the GA solution by augmenting the GA FF by appropriate penalty terms for each violated functional constraint. Constraints on the control variables (8) are automatically satisfied by the selected GA encoding/decoding scheme.

Therefore, the GA FF is formed as follows Equation (10 and 11):

$$FF = \frac{A}{\sum_{i=1}^{Ng} Fi(PGi) + \sum_{i=1}^{Nc} \omega_j \cdot Pen_j} \tag{10}$$

$$Pen_j = |h_j(x, u)| \cdot H(h_j(x, u)) \tag{11}$$

Where:

- FF = Fitness function
- A = Constant
- Fi(PGi) = Fuel cost of unit i
- H(.) = Heaviside (step) function
- NG = Number of units
- Nc = Number of functional operating constraints

5. ENHANCED GENETIC ALGORITHM (EGA)

In the EGA, (Goyal and Singh, 2012) the application of the basic genetic operators (parent selection, crossover

and mutation) the advanced and problem-specific operators are applied to produce the new generation.

All chromosomes in the initial population are created at random (every bit in the chromosome has equal probability of being switched ON or OFF). Due to the decoding process selected, the corresponding control variables of the initial population satisfy their upper–lower bound or discrete value constraints. However, the initial population candidate solutions may not satisfy the functional operating constraints or even the load flow constraints since the random, within limits, selection of the control variables may lead to load flow divergence.

Population statistics computed for the new generation include maximum, minimum and average fitness values and the 90% percentile.

Population statistics are then used to adaptively change the crossover and mutation probabilities. If premature convergence is detected the mutation probability is increased and the crossover probability is decreased. The contrary happens in the case of high population diversity.

5.1. Advanced and Problem-Specific Genetic Operators

One of the most important issues in the genetic evolution is the effective rearrangement of the genotype information. In the SGA, crossover is the main genetic operator responsible or the exploitation of information while mutation brings new non-existent bit structures. It is widely recognized that the SGA scheme is capable of locating the neighbourhood of the optimal or near-optimal solutions, but in general, requires a large number of generations to converge. This problem becomes more intense for large-scale optimization problems with difficult search spaces and lengthy chromosomes, where the possibility for the SGA to get trapped in local optimal increases and the convergence speed of the SGA decreases.

At this point, a suitable combination of the basic, advanced and problem-specific genetic operators must be introduced in order to enhance the performance of the GA. Advanced and problem-specific genetic operators usually combine local search techniques and expertise derived from the nature of the problem.

A set of advanced and problem-specific genetic operators has been added to the SGA in order to increase its convergence speed and improve the quality of solutions. Our interest was focused on constructing simple yet powerful enhanced genetic operators that effectively explore the problem search space. The advanced features included in our GA implementation are as follows.

5.2. Fitness Scaling

In order to avoid early domination of extraordinary strings and to encourage a healthy competition among equals, a scaling of the fitness of the population is necessary. In our approach, the fitness is scaled by a linear transformation.

5.3. Elitism

Elitism ensures that the best solution found thus far is never lost when moving from one generation to another. The best solution of each generation replaces a randomly selected chromosome in the new generation.

5.4. Hill Climbing

In order to increase the GA search speed at smooth areas of the search space a hill-climbing operator is introduced, which perturbs a randomly selected control variable. The modified chromosome is accepted if there is an increase in FF value; otherwise, the old chromosome remains unchanged. This operator is applied only to the best chromosome (elite) of every generation.

In addition to the above advanced features, which are called “advanced” despite their wide use in most recent GA implementations to distinguish between the SGA and our EGA, operators specific to the OPF problem have been added.

All problem-specific operators introduce random modification to all chromosomes of a new generation. If the modified chromosome proves to have better fitness, it replaces the original one in the new population. Otherwise, the original chromosome is retained in the new population. All problem-specific operators are applied with a probability of 0.2. The following problem-specific operators have been used.

5.5. Gene Swap Operator (GSO)

This operator randomly selects two genes in a chromosome and swaps their values as shown in **Fig. 2**. This operator swaps the active power output of two units, the voltage magnitude of two-generation buses. Swapping among different types of control variables is not allowed.

5.6. Gene Cross-Swap Operator (GCSO)

The GCSO is a variant of the GSO. It randomly selects two different chromosomes from the population and two genes, one from every selected chromosome and swaps their values as shown in **Fig. 3**. While crossover exchanges information between high-fit chromosomes, the GCSO searches for alternative alleles, exploiting information stored even in low-fit strings.

5.7. Gene Copy Operator (GCO)

This operator randomly selects one gene in a chromosome and with equal probability copies its value to the predecessor or the successor gene of the same control type as shown in **Fig. 4**. This operator has been introduced in order to force consecutive controls (e.g., identical units on the same bus) to operate at the same output level.

5.8. Gene Inverse Operator (GIO)

This operator acts like a sophisticated mutation operator. It randomly selects one gene in a chromosome and inverses its bit-values from one to zero and vice versa as shown in **Fig. 5**. The GIO searches for bit structures of improved performance, exploits new areas of the search space far away from the current solution and retains the diversity of the population.

5.9. Gene Max-Min Operator (GMMO)

The GMMO tries to identify binding control variable upper/lower limit constraints. It selects a random gene in a chromosome and, with the same probability (0.5), fills its area with 1s or 0s as shown in **Fig. 6**.

The step by step procedure for the Enhanced Genetic Algorithm is as follows

- Input of the data: vlb, vub, PC, Pm, the function of adaptation and size of the population
- To choose arbitrary the initial population
- To decode the chains to calculate the value of the function to be optimized. For that, it is enough to inject the values of chains decoded in the function
- To use the three following operators:
 - Reproduction
 - Crossover
 - Mutation
- To use the three following operators:
 - Reproduction
 - Crossover
 - Mutation
- Advanced and problem specific operators
 - Hill Climbing
 - Gene swap operator
 - Gene cross-swap operator
 - Gene copy operator
 - Gene Inverse Operator
 - Gene max-min operator
- If the convergence of GAs is reached we print the optimal values and stop;
- else go to the second step

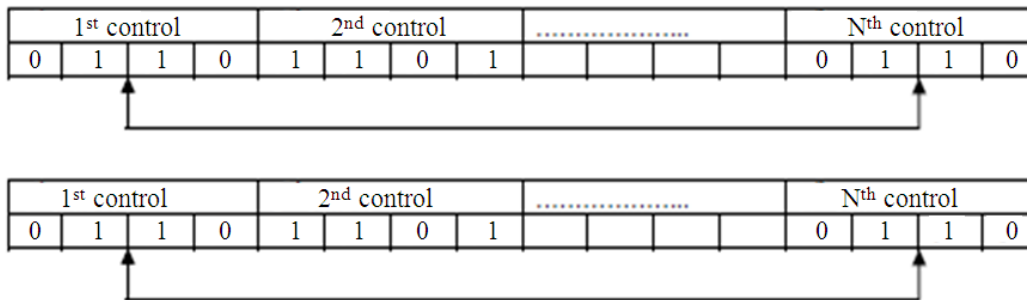


Fig. 2. Gene swap operator

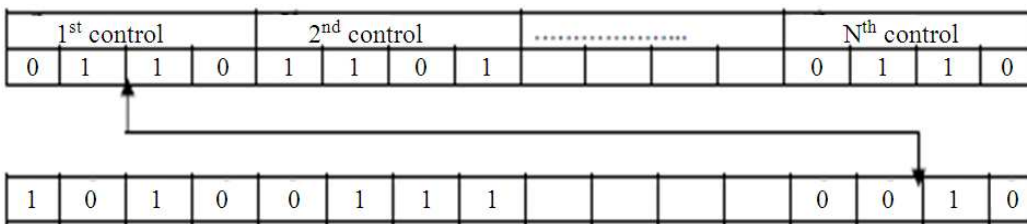


Fig. 3. Gene cross-swap operator

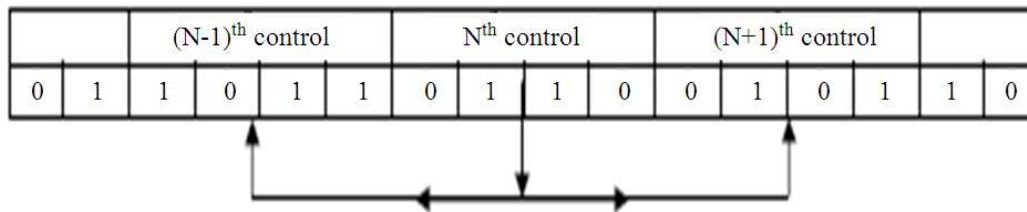


Fig. 4. Gene copy operator

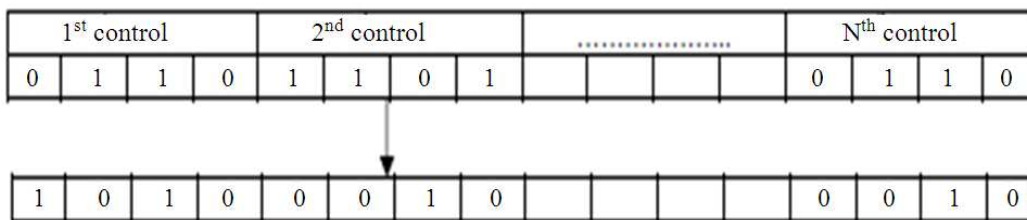


Fig. 5. Gene inverse operator

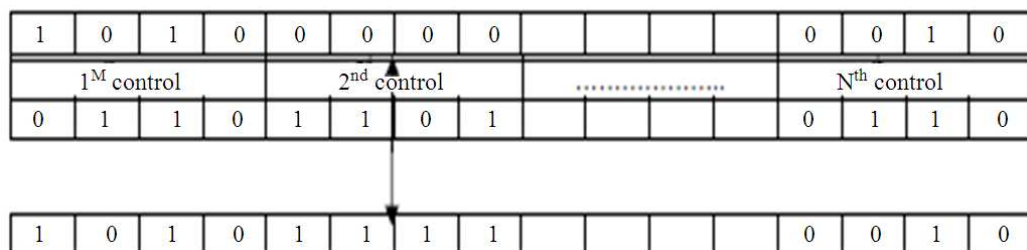


Fig. 6. Gene max-min operator

6. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a population-based stochastic optimization technique (Kiran *et al.*, 2012) developed by Eberhart and Kennedy (1995), inspired by the social behaviour of bird flocking or fish schooling. These phenomena can also be observed on insect colonies, e.g., bees. It is applicable to solving a number of problems where local methods fail or their usage is ineffective, as in this case. One of the most important features of PSO is the ability of optimizing large complex multi-criteria combinatorial problems where the problem with the design of criteria function occurs, for example, it is hard to derive or is not continuous.

PSO however does not need this as it only requires the evaluation of each solution by the fitness function depending on the set of optimized parameters. This function is also used by GA and so is the idea of the initialization of parameter setup as a random generation. The main advantage of PSO compared to GA is the simpler method of providing new solutions based only on two variables-velocity and position related by two linear equations. Each possible solution are represented by a particle, which flies through the searched space, which is limited by restrictive maximum and minimum values, toward the current optimal position. The particle has its direction and speed of movement (velocity) but it can also randomly decide to move to the best position of all positions or to its own best position. Each particle holds information about its own position (which represents one potential solution), the velocity and the position with the best fitness function it ever has flown through.

The flow sequence of the Particle Swarm Optimization Algorithm (Poli *et al.*, 2007) is illustrated as follows:

Algorithm PSO

Begin

Generate random population of N solutions(particles);

For each individual $I \in N$ calculate fitness (i);

Initialize the value of the weight factor ω ;

For each particle;

Set pBest as the best position of particle i ;

If fitness (i) is better than pBest;

pBest(i)=fitness (i);

End;

Set gBest as the best fitness of all particles;

For each particle;

Calculate particle velocity according to Equation (12);

Update particle position according to Equation (13);

End;

Update the value of the weight factor ω (option);

Check if termination=true;

End

6.1. Implementation

The program was implemented in the Matlab environment. The position here represents one potential solution, the velocity shows the trend of this particle and both parameters are represented by a vector in the program implementation. The particles were coded by natural numbers. The position of each element in the vector space represents the number of the node in which a shunt capacitor should be placed whose value designates the capacitor type. The whole set of particles at a time is called the population. The subset made of newly born particles is called the generation.

The first generation of particles is produced with random position and velocity. Particle velocity is checked whether it is within the limits. The top speed can be different for each unit of velocity vector. If the velocity component exceeds the maximum allowed value, then it is set to the top value. After this correction, the solution is evaluated by the fitness function. The fitness function plays a key role in the program; therefore it is necessary to describe it in more details.

6.2. Fitness and penalization functions

The fitness function evaluates the quality of solutions and it incorporates numerous parameters, such as the capital cost of capacitors, expenses covering the power losses in the network per year and function γ . The power losses are calculated by steady state analysis of the network. The output of the fitness function is total yearly operational costs of the network. The lower the fitness function value, the better the solution. The fitness function is calculated by the following equation:

$$FF = \frac{A}{\sum_{i=1}^{Ng} Fi(PGi) + \sum_{j=1}^{Nc} \omega_j \cdot Pen_j} \quad (12)$$

Where:

FF = Fitness function

A = Constant

Fi(PGi) = Fuel cost of unit i

H(.) = Heaviside (step) function

NG = Number of units

Nc = Number of functional operating constraints (2):

$$Pen_j = |h_j(x, u)| \cdot H(h_j(x, u)) \tag{13}$$

6.3. Next population

After evaluation, the solutions can be sorted with respect to their fitness functions and it is possible to develop a new generation, as can be seen in **Fig. 2**. The first cycle ends after the creation of a new velocity vector and the calculation of the new position of the particular particle.

The new vector of velocity is calculated by the formula Equation (14):

$$\bar{v} = \bar{v}_0 + c_1 \cdot n_1 \cdot (\bar{P}_{best} - \bar{P}_{pos}) + c_2 \cdot n_2 \cdot (\bar{g}_{best} - \bar{g}_{pos}) \tag{14}$$

where, \bar{v} denotes the new vector of velocity, \bar{v}_0 is the original vector of velocity, c_1 and c_2 are the constants which are set to the weight of differences of positions, n_1 and n_2 are the random variables, \bar{P}_{best} is the best position of particle, \bar{P}_{pos} is the current position of particle and best \bar{g}_{best} is the best position of all particles.

The new position is determined by the formula Equation (15):

$$\bar{P}_{best} = \bar{P}_{pos} + \bar{v} \tag{15}$$

6.4. Border

Each particle should be kept in a confined space corresponding to the parameter limitations. This problem is solved in this program by one of four methods. In the first case, the particle arriving in the forbidden area returns to its previous position. In the second case, the particle is held on the border. In the third case, the particle is bounced back to the allowed space. Bouncing back to the allowed space can be perfect or imperfect. Regarding the imperfect bounce, it is possible for the particle to end up in a random position. In the fourth case, the particle can fly through the forbidden area back to the allowed space, but on the other side of the allowed space. This approach can be used in the case of a very specific limited space (**Fig. 7**).

6.5. Parallel operation

One of the advantages of PSO utilization is the possibility to introduce parallel operations with mutual coupling, which enables searching in a larger area of

feasible solutions and thereby finding the optimum solution more quickly. Parallel operation means that instead of a single branch of evolution several branches are created. These branches influence one another during the evolution only after a given number of generations when the temporary best solution of all the branches is transferred to the other branches. Thus the evolution of the branches is independent, but they can also use the results of the other branches. This modification limits a potential deadlock of the algorithm in the local minimum.

7. TEST RESULTS

A MATLAB coding is developed for each algorithm inter linked with the conventional Newton Raphson's method for load flow study. IEEE 14-bus system is taken to verify the effective operation of the algorithm. The **Fig. 8** shows the line diagram of IEEE 14-bus system.

The total population size in each algorithm is selected as 30, the mutation probability as 0.01 and crossover probability as 1.0.

From the coding results obtained with genetic algorithm it is found that each time the coding is executed a new result is obtained as tabulated in **Table 3**. From the observation made from the **Table 3** we could say that with genetic algorithm analysis the use of TCSC controller with rating of 1.0 to -1.0 p.u at the line connecting the bus 4 to bus 9.

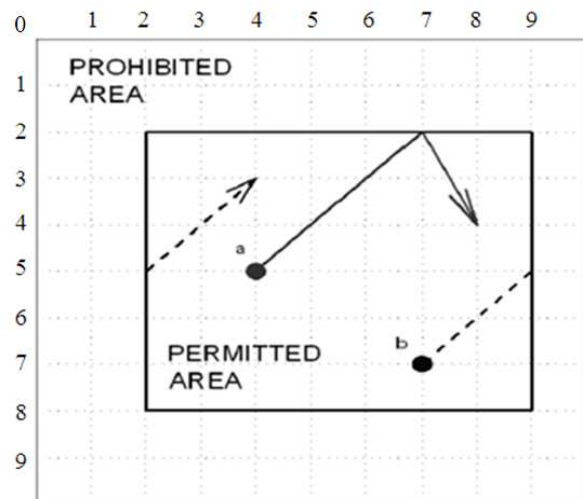


Fig. 7. Border of the space under examination-particle is returned to allowed space by (a) perfect bounce; (b) flight through prohibited area

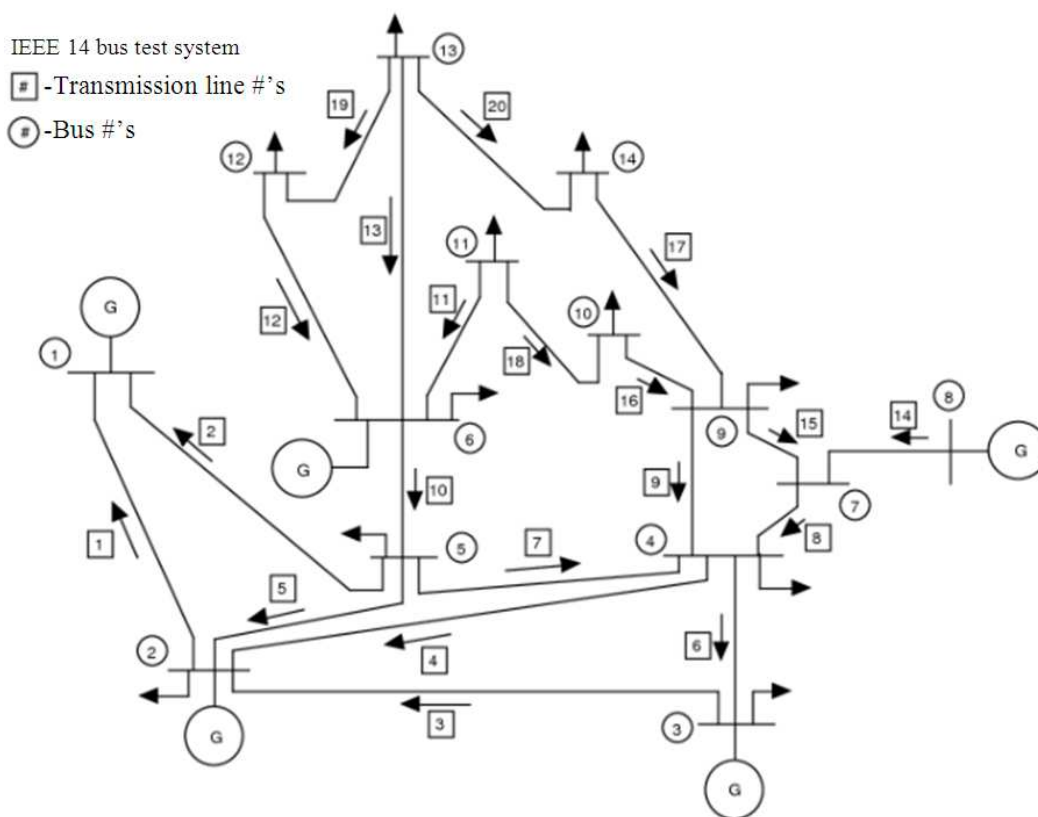


Fig. 8. Line diagram of IEEE 14-bus system

Table 3. Result obtained for simple genetic algorithm

SI	Device		Bus	
	Type	Rating (p.u)	nl	nr
1	UPFC	-0.6086	9	14
2	TCSC	-0.9657	4	9
3	UPFC	-0.3852	4	7
4	UPFC	0.6015	3	4
5	TCSC	0.7654	4	9

Table 4. Result obtained for enhanced genetic algorithm

SI	Controller		Bus	
	Type	Rating (p.u)	nl	nr
1	TCSC	-0.2766	2	3
2	TCSC	-1.0000	4	7
3	UPFC	0.9899	1	5
4	UPFC	-0.2212	1	5
5	TCSC	0.5297	5	6

Table 5. Result obtained for particle swarm algorithm

SI	Controller		Bus	
	Type	Rating (p.u)	nl	nr
1	UPFC	-0.0496	5	6
2	UPFC	-0.2579	4	9
3	UPFC	-0.1543	5	6
4	TCSC	-0.1129	5	6
5	UPFC	-0.9757	4	9

From the result obtained it is found that the use of UPFC controller will be more efficient with a rating of 1.0 to -0.3 p.u at the line connecting the bus 1 to bus 5. Though the TCSC controller is repeated more number of times than UPFC controller the bus location for TCSC controller is different in each time when TCSC controller is selected **Table 4.**

Table 5 shows the coding result obtained for particle swarm algorithm is used. The controller selected by this algorithm is UPFC with a rating ranging from 0.0 to -1.0 p.u at the line connecting bus 5 to bus 6.

8. CONCLUSION

The three optimization techniques were coded to present the optimal solution to the power flow problem in power system and to find the rating and location of FACTS devices. In each of the method the power injection model of FACTS device is incorporated with the conventional AC optimal power flow.

On comparing the results obtained from each optimization techniques it can be said that the use of UPFC controller at the line connecting the bus 5 to bus 6 with the rating of 1.0p.u to -1.0p.u will give a better feasible solution with transmission of maximum power transmission in the power system.

However, it can also increase the controllability and feasibility of the system and provide a wider operating margin and higher voltage stability with higher reserve capacity. In these optimization methods, all the techniques can effectively find the optimal setting of the control parameters using the conventional load flow method.

The further scope of this study is to extend the work to find the location of the device in the transmission line entire stretch and also to find the effect of the device in power system.

9. REFERENCES

- Eberhart, R. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Oct. 4-6, IEEE Xplore Press, Nagoya, pp: 39-43. DOI: 10.1109/MHS.1995.494215
- Goyal, S.K. and M. Singh, 2012. Enhanced genetic algorithm based load balancing in grid. *Int. J. Comput. Sci.*, 9: 260-266.
- Kiran, S.H., C. Subramani, S.S. Dash, M. Arunbhaskar and M. Jagadeeshkumar, 2011. Particle swarm optimization algorithm to find the location of facts controllers for a transmission line. Proceedings of the International Conference on Process Automation Control and Computing, Jul. 20-22, IEEE Xplore Press, Coimbatore, pp: 1-5. DOI: 10.1109/PACC.2011.5978856
- Kiran, S.H., C. Subramani, S.S. Dash, M. Arunbhaskar and M. Jagadeeshkumar, 2012. Particle swarm optimization algorithm to find the location of facts controllers for a transmission line. *Adv. Intell. Soft Comput.*, 132: 861-868. DOI: 10.1007/978-3-642-27443-5_99
- Poli, R., J. Kennedy and T. Blackwell, 2007. Particle swarm optimization an overview. *Swarm Intell.*, 1: 33-57. DOI: 10.1007/s11721-007-0002-0
- Popov, A. and Hamburg, 2008. Genetic algorithm for optimization.
- Subramani, C., S.S. Dash, V. Kumar and H. Kiran, 2012. Implementation of line stability index for contingency analysis and screening in power system. *J. Comput. Sci.*, 8: 585-590. DOI: 10.3844/jcssp.2012.585.590