ISSN: 2229-712X

Design and analysis of optimal power flow for power system using lagrangian relaxation technique

S. Maheswari and C. Vijayalakshmi
Department of Mathematics, Sathyabama University, Chennai – 119.

ARTICLE INFO

ABSTRACT

This paper mainly deals with the solution of Optimal Power Flow (OPF) problem which can be formulated as a Quadratic Programming (QP) model and then decomposed by Lagrangian Relaxation (LR) method.The objective of this model is to minimize the total cost of real power generation. Many researchers have discussed the solution for OPF by using different methodologies like Newton Raphson (NR) method, Particle Swarm Optimization (PSO) method, Genetic Algorithm (GA), Artificial Intelligence (AI) method, Interior Point (IP) method, Differential Evolution (DE) algorithm. In this paper the proposed methodology is compared with the other methods like Particle Swarm Optimization method, Genetic Algorithm and Differential Evolution algorithm. These methods have been tested through the results of IEEE 30 bus system. The optimum generation cost is minimized for OPF using Lagrangian Relaxation method.

Introduction

The Optimal Power Flow (OPF) was discussed by Carpentier in 1962. OPF has been widely used in power system and management. After restructuring the electricity sector OPF is a tool which is used to minimize the power production cost by adjusting the power system control variables. The objective of OPF is to minimize the generation cost and / or transmission losses. The optimal operation of power system is to determine the power schedule so that the total cost of operation is minimized with respect to operating constraints. The constraints involved are the physical laws governing the power generation – transmission systems and operating limitations of the equipment. The power flow study in a power system is required for planning the operation of power systems with respect to existing conditions and its future expansion. J.B. Gupta [1] has introduced the power flow for the active and reactive power.
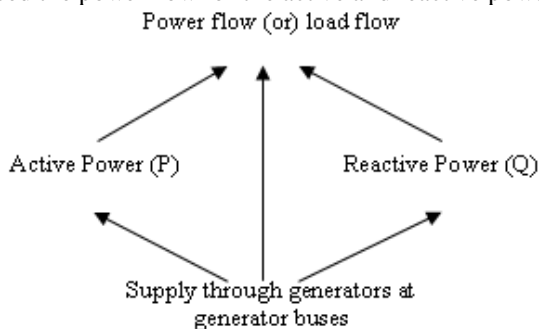

Fig:1 Optimization of Power Flow

Active Power (P)
Active power is drawn by loads from load buses.
Reactive Power (Q)
Reactive power is supplied (or) drawn from the load buses by shunt compensation elements like shunt capacitors, reactor elements, static VAR system.
The load flow studies in essential for future system expansion to meet the increased load demand.

Operation of the power grid at steady state is one of the most fundamental requirement of proper operation of a power system. The steady state operation of the power network is principally governed by the system voltage at the two ends, the transfer reactance of the line and the power angle between the two buses.
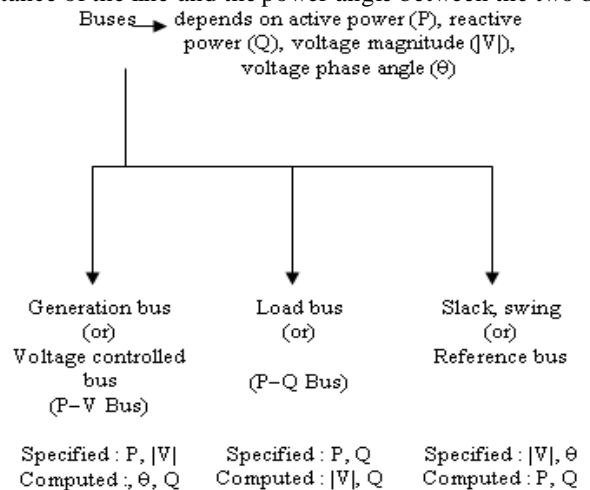

Fig:2 classification of Buses

Literature Review

T.S. Chung et al. [2] has discussed recursive linear programming which minimizing line losses and finding the optimal capacitor allocation in a distribution system. E. Lobatu et al. [3] proposed LP based OPF for minimization of transmission losses and generator reactive margins of the Spanish power system.

S. Chen et al. [4] have designed a new algorithm based on Newton-Raphson (NR) method in order to solve emission dispatch in real tune. X. Tong et al. [5] presented semi smooth Newton-type algorithms for solving OPF problems. These algorithms separated inequality and bounded constraints.

J.A. Momoh [6] has discussed the extension of basic Kuhn-Tucker conditions and generalized quadratic-based model for

Tele:
E-mail addresses: rohithkumar2007@gmail.com

OPF. N. Grudinin [7] has designed a reactive power optimization model which is based on successive QP (SQP) methods. These methods used to test 30 bus and 278 bus systems.Feasibility, convergence and optimal. Execution time is reduced. SQP methods provide more fast and reliable optimization.

D.Pudjianto et al. [8] used LP and NLP based reactive OPF for allocating reactive power among competing generators in a deregulated environment. G.L. Torres et al. [9] proposed the methods to calculated the price of reactive power support service in a multi-area power system. Methods which are based on Cost Benefit Analysis (CBA) and linear convex network flow programming.

LP method calculated the over all cost associated with the system reactive requirement. It gives reasonably accurate. NLP gives a faster computation speed and accuracy for the solution. The reactive power support benefits with respect to power delivery increases of tie lines. Generators individual commitment vary. The convergence could not be guaranteed for every condition.

Ding Xiaoying et al. [10] have discussed an Interior Point Branch and Cut Method (IPBCM) to solve decoupled OPF problem. The Modern Interior Point Algorithm (MIPA) is used to solve Active Power Sub Optimal Problem (APSOP) and use IPBCM to iteratively solve linearization of Reactive Power Sub Optimal Problem (RPSOP). Wei Yan et al. [11] presented the solution of the optimal reactive power flow (ORPF) problem by the Predictor Corrector Primal Dual Interior Point Method (PCPDIPM). ORPF was designed as a model in rectangular formal the Hessian matrices in this model are constants, it has been evaluated only once in the entire optimal process.

The variables and constraints of RPSOP are less than that of original OPF problem, which gives the fast calculation speed.

N.I. Santoso et al. [12] have discussed a two-stage Artificial Neural Network (ANN) to control in real time the multi tap capacitors installed on a non conforming load profile such that the system losses are minimized. Walters et al. [13] applied a genetic algorithm (GA) to solve an economic dispatch problem for valve point discontinuities. T.C. Chung et al. [14] have proposed a Hybrid Genetic Algorithm (GA) method to solve OPF in corporating FACTS devices. GA is integrated with conventional OPF to select the best control parameters to minimize the total generation fuel cost and keep the power flows with in the security limits. It converged in a few iterations.

H. Yoshida et al. [15] have discussed a particle swarm optimization (PSO) for reactive power and voltage / VAR control (CCV) considering voltage security assessment. It determined an online VVC strategy with continuous and discrete control variables, Cui Ru Wang et al. [16] presented a modified particle swarm optimization (MPSO) algorithm to solve economic dispatch problem. I.K. Yu et al. [17] have proposed a novel cooperative agents approach, Ant colony search algorithm (ACSA) based scheme, for solving a short-term generation scheduling problem of thermal power systems.

P. Somasundaram et al. [18] have discussed an algorithm for solving security constrained optimal power flow problem through the application of EP. The controllable system quantities in the base case state are optimized to minimize some defined objective function subject to the base-case operating constraints fitness function converges smoothly without any oscillations.

Many researchers have discussed the solution of OPF by different optimization techniques. In this paper, the proposed model and methodology gives the appropriate schedule and the optimized value for the operation of power system. The model is designed with respect to various constraints and the objective is mainly to include slack buses that reduce the losses which minimize the total generation cost.

Optimization techniques for the solution of OPF
## Genetic Algorithms

GAs are general purpose optimization algorithms 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.

Genetic evolution takes place by means of three basic genetics operators:
1) parent selection;
2) crossover;
3) 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.

Simple Genetic Algorithm (SGA) Flow Chart
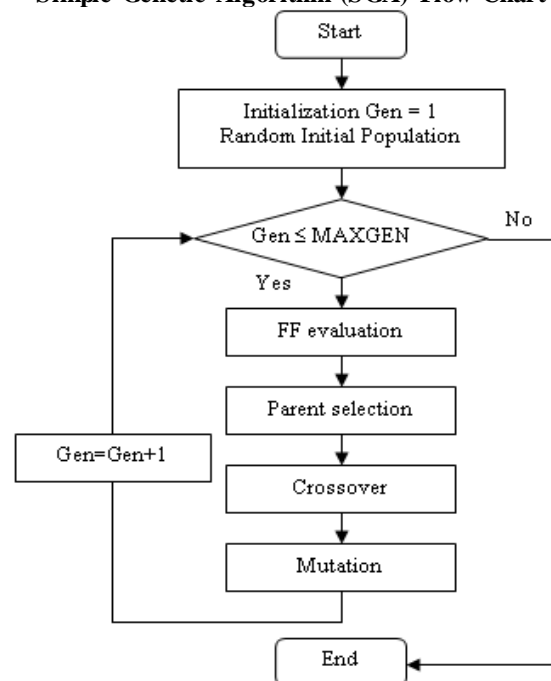
Crossover is an extremely important operator for the GA. It is responsible for the structure recombination (information exchange between mating chromosomes) and the convergence speed of the GA and is usually applied with high probability (0.6–0.9). The chromosomes of the two parents selected are combined 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 FF evaluation and genetic evolution take part in an iterative procedure, which ends when a maximum number of generations is reached, as shown in Fig. 1. When applying GAs to solve a particular optimization problem (OPF in our case), two main issues must be addressed:

1) 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).

Genetic Algorithm Solution to Optimal Power Flow
A. Encoding

There are four chromosome regions (one for each set of control variables), namely, 1) $\hat{P}_G$; 2) U_G; 3) t; and 4) b_{SH}. Encoding is performed using different gene-lengths for each set of control variables, depending on the desired accuracy. The decoding of a chromosome to the problem physical variables is performed as follows:

1) continuous controls taking values in the interval $[u_i^{min}, u_i^{max}]$

$$u_i = u_i^{min} + (u_i^{max} - u_i^{min}) \frac{k}{2}$$ (a)

2) discrete controls taking M values $u_i^1, u_i^2, \dots u_i^M$

$$u_i = u_i^{(l)} \quad with \quad l = \frac{M}{2}$$

and

$$1 \le M \le 2^{N}$$ (b)

where k is the decimal number to which the binary number in a gene is decoded and N_{u_i} is the gene length (number of bits) used for encoding control variable u_i.

Fitness Function (FF)

GAs are 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.

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 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 (4) are automatically satisfied by the selected GA encoding/decoding scheme (a) and (b).

Therefore, the GA FF is formed as follows:

$$FF = \frac{A}{\sum_{i=1}^{N_G} F_i(P_{Gi}) + \sum_{j=1}^{N_C} w_j P_j}$$ (c)

$$P_j = |h_j(x,u) \cdot H(h_j(x,u))|$$ (d)

where

FF	fitness function;
A	constant;
$F_i(P_{Gi})$	fuel cost function of unit (in our case, a quadratic

function);
w_j weighting factor of functional operating constraint j;
Pen_j penalty function for functional operating constraint j;
$h_j(x, u)$ violation of jth functional operating constraint, if positive;
$H(\Box)$ Heaviside (step) function;
N_G number of units;
N_C number of functional operating constraints.

Given a candidate solution to the problem, represented by a chromosome, the FF is computed as follows.

Step 1) Decode the chromosome to determine the actual control variables, using (a) and (b). The computed control vector satisfies, by design, constraints.

Step 2) Solve the power flow to compute the state vector, x.

Step 3) Determine the violated functional constraints and compute associated penalty functions (d).

Step 4) Compute the FF using (c).

In Step 2, a simple Fast Decoupled Load Flow (FDLF) is used with no PV-PQ bus-type switching, since generator reactive capabilities are incorporated in the functional operating constraints and no local control adjustments, such as tap and switchable shunts, since the settings of these controls are determined by the GA. Therefore, only a few load flow iterations are required for convergence. The FDLF and matrices are formed and factorized only once in the beginning and the effect of the changes of shunt admittances on the matrix is neglected. In case that, due to the random (yet within limits) initial selection of the control variables, the load flow does not converge within a predefined number of iterations (set to 8), large penalty terms, proportional to the maximum active/reactive power mismatch, are added.

Particle Swarm Optimization Method

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberharth and Dr. Kennedy in 1995. In PSO, the potential solutions called particles, fly through the problem space by the current optimum particles.

Compromise of a given particle is formalized by the following equations:

$$\vec{v}_{k+1} = \vec{a}\vec{v}_k + \vec{b}_1\vec{r}_1(\vec{p}_1 - \vec{x}_k) + \vec{b}_2\vec{r}_2(\vec{p}_2 - \vec{x}_k)$$
$$\vec{x}_{k+1} = \vec{x}_k + \vec{v}_{k+1}$$ (e)

where:

$\vec{v}_{k+1}$ the current velocity.

$\vec{a}$: The inertia weighting function.

$\vec{v}_k$: the previous velocity.

$\vec{b}_1, \vec{b}_2$: the cognitive and the social parameters, respectively.

$\vec{r}_1, \vec{r}_2$: random numbers uniformly distributed within [0, 1].

$\vec{p}_1$: the best previous position of the kth particle.

$\vec{p}_2$: the global best in the kth swarm.

$\vec{x}_{k+1}$: the current position.

$\vec{x}_k$: the previous position.

The first part of equation (e) is the inertia velocity of particle, which reflects the memory behavior of particle; the second part is cognition part, which represents the private

thinking of the particle itself; the third part is the social part, which shows the particle's behavior system from the experience of other particles in the population. The particles find the optimal solution by cooperation and competition among the particles.

Using the above equation, a certain velocity, that gradually gets close to $\vec{p}_1$ and $\vec{p}_2$, can be calculated. The position of each particle (searching point in the solution space) can be modified.

The cost function is defined as:

$$F=\sum_{i=1}^{ng} C_i(P_{ig}) = a_i + b_i P_{ig} + c_i P_{ig}^2 \qquad (f)$$

$$P_{ig}^{min} \leq P_{ig} \leq P_{ig}^{max}$$

Minimize F is equivalent to getting a maximum fitness value in the searching process. The particle that has lower cost function should be assigned a larger fitness value. The objective of OPF has to be changed to the maximization of fitness to be used as follows:

$$fitness = \begin{cases} (F_{max} - F)/F_{max} \\ 0 \qquad oth \end{cases} \qquad (g)$$

The PSO algorithm applied to OPF can be described in the following steps:

Step 1: Input parameters of system, and specify the lower and upper boundaries of each variable.

Step 2: Initialize randomly the particles of the population.

Step 3: Calculate the evaluation value of each particle using the objective function.

Step 4: Calculate the fitness value of objective function of each particle using (6), $\vec{p}_1$ is set as the kth particle's initial position; $\vec{p}_2$ is set as the best one of $\vec{p}_1$, and the current evolution is t = 1.

Step 5: Initialize learning factor $\vec{b}_1, \vec{b}_2$, inertia weight $\vec{a}$ and the initial velocity $\vec{v}_1$

Step 6: Modify the velocity $\vec{v}$ of each particle according to (e).

Step 7: Modify the position of each particle. If a particle violates its position limits in any dimension, set its position at the proper limits. Calculate each particle's new fitness, if it is better than the previous $\vec{p}_2$, the current value is set to be $\vec{p}_2$.

Step 8: To each particles of the population, employ the Newton-Raphson method to calculate power flow and the transmission loss.

Step 9: Update the time counter t = t + 1.

Step 10: If one of the stopping criteria is satisfied then go to step 11. Otherwise go to step 7.

Step 11: The particle that generates the latest $\vec{p}_2$ is the Pareto optimal value.

DE Optimization Method
Initialization

The first step in the DE optimization process is to create an initial population of candidate solutions by assigning random values to each decision parameter of each individual of the population. Such values must lie inside the feasible bounds of the decision variable and can be generated. In case a preliminary solution is available, adding normally distributed random deviations to the nominal solution often generates the initial population.

Flow Chart for PSO
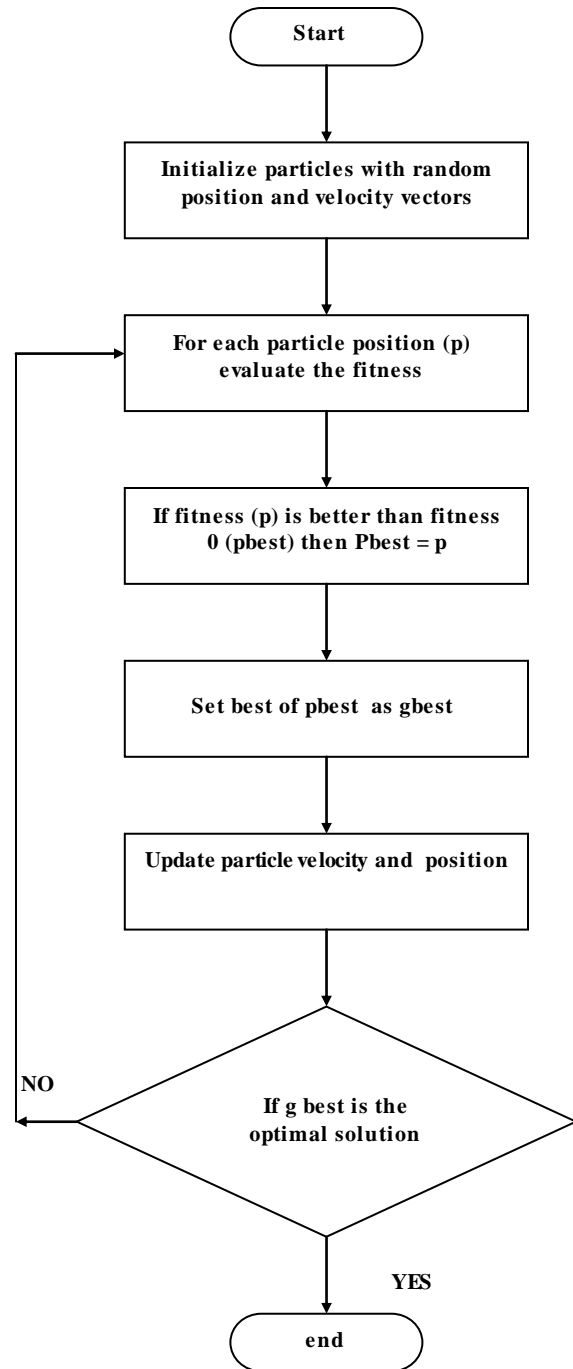
$$Y_j^0 = Y_j^{min} + \sigma_j (Y_j^{max} - Y_j^{min}) \qquad (h)$$

$i=1, 2,\ldots\ldots Np, \qquad j=1, 2,\ldots\ldots D$

where Y_j^{min} and Y_j^{max} are respectively, the lower and upper bound of the j th decision parameter and σ_j is a uniformly distributed random number within [0,1] generated anew for each value of j.

Mutation

After the population is initialized, this evolves through the operators of mutation, cross over and selection. For crossover and mutation different types of strategies are in use. Basic scheme is explained here elaborately. The mutation operator is in charge of introducing new parameters into the population. To achieve this, the mutation operator creates mutant vectors by perturbing a randomly selected vector (Y_a) with the difference of two other randomly selected vectors (Y_b and Y_c) according Eq. (i). All of these vectors must be different from each other, requiring the population to be of at least four individuals to

satisfy this condition. To control the perturbation and improve convergence, the difference vector is scaled by a user defined constant in the range [0, 1.2]. This constant is commonly known as the scaling constant (S*)*.

$$Y_i^{(G)} = Y_a^{(G)} + S(Y_b^{(G)} - Y_c^{(G)}) \quad i=1, 2,\ldots\ldots Np \qquad (i)$$

where Y_a, Y_b and Y_c are randomly chosen vectors $\in \{i = 1, 2,\ldots\ldots Np\}$ and a $\square$ b $\square$ c $\square$ i.

Y_a, Y_b and Y_c are generated a new for each parent vector, *S* is the scaling constant. For certain problems, it is considered a = i.

Crossover

The crossover operator creates the trial vectors, which are used in the selection process. A trail vector is a combination of a mutant vector and a parent (target) vector based on different distributions like uniform distribution, binomial distribution; exponential distribution is generated in the range [0, 1] and compared against a user defined constant referred to as the crossover constant. If the value of the random number is less or equal than the value of the crossover constant, the parameter will come from the mutant vector, otherwise the parameter comes from the parent vector as given in Eq. (j).

The crossover operation maintains diversity in the population, preventing local minima convergence. The crossover constant (*CR*) must be in the range of [0, 1]. A crossover constant of one means the trial vector will be composed entirely of mutant vector parameters. A crossover constant near zero results in more probability of having parameters from the target vector in the trial vector. A randomly chosen parameter from the mutant vector is always selected to ensure that the trail vector gets at least one parameter from the mutant vector even if the crossover constant is set to zero.

$$X_{i,j}^{''(G)} = \begin{cases} X_{i,j}^{'(G)} & if\ \square_j' \le C_R\ or\ j=q \\ X_{i,j}^{'(G)} & otherwise \end{cases}$$

where $i=1, 2,\ldots\ldots Np$, $j=1, 2,\ldots\ldots D$ $\qquad (j)$

q is a randomly chosen index $\in \{i=1, 2,\ldots\ldots Np\}$ that guarantees that the trial vector gets at least one parameter from the mutant vector; $\eta\square_j$ is a uniformly distributed random number within [0, 1) generated anew for each value of j. $X_{i,j}^{(G)}$ is the parent (target) vector, $X_{i,j}^{'(G)}$ the mutant vector and $X_{i,j}^{''(G)}$ the trial vector.

Selection

The selection operator chooses the vectors that are going to compose the population in the next generation. This operator compares the fitness of the trial vector and fitness of the corresponding target vector, and selects the one that performs better.

$$Y_i^{(G+1)} = \begin{cases} Y_i^{''(G)} & if\ f(Y^{''(G)}) \le f(Y^{(G)}) \\ Y_i^{'(G)} & otherwise \end{cases}$$

$i=1, 2,\ldots\ldots Np$ $\qquad (k)$

The selection process is repeated for each pair of target/ trail vector until the population for the next generation is complete.

Application of DE to OPF

Differential Evolution has been applied to problems from several areas. Some power engineering problems have been solved with DE including: Distribution systems capacitors placement, harmonics voltage distribution reduction and passive shunt harmonic filter planning. DE has also been used in the design of filters, neural network learning, fuzzy logic application, and optimal control problems, among others.

The objective function of OPF

$$F_{cost} = \sum_{i=1}^{Ng} (a_i P_{Gi}^2 + b_i P_{Gi} + c_i) \quad \$/Hr \qquad (l)$$

subjected to the constraints $g(x,u) = 0$, $h(x,u) \le 0$ where g is the equality constraints and represent typical load flow equations. *h* is the system operating constraints

Dependent Variables

X is the vector of dependent variables consisting of slack bus power P_{G1}, load bus voltages V_L, generator reactive power outputs Q_G, and transmission line loadings S_l. Hence, *X* can be expressed as

$$X^T = [P_G, V_L, Q_G, S_l]$$

i.e.,

$$X^T = [P_{G1}, V_{L1}\ldots V_{LNl}, Q_{G1}\ldots Q_{GNg}, S_{l1}\ldots S_{lNl}] \qquad (m)$$

where N_l, Ng, Npq are number of load buses, number of generators, and number of transmission lines, respectively.

Independent Variables

U is the vector of independent variables consisting of generator voltages *V G*, generator real power outputs *P G*, except at the slack bus 1 *P G*, and transformer tap settings T. Hence, *U* can be expressed as

$U= [V_G, P_G, T]$

i.e.,

$u^T = [V_{G1},\ldots\ldots V_{GNg}, P_{G2}\ldots\ldots P_{GNg}, T_1\ldots\ldots T_{Nt}] \qquad (n)$

Initialization

The first step in this algorithm is to create an initial population. All the independent variables $[V_G, P_G, T]$ have to be generated according to formula (3), where each independent parameter of each individual in the population is assigned a value inside the given feasible region of the generator. This creates parent vectors of independent variables for the first generation. As they have created within their limits, they readily satisfy the corresponding inequality constraints. To find dependent variables $X^T= [P_{G1}, V_L, Q_G, S_l]$, corresponding to each individual, Newton-Raphson power flow solution is implemented.

After getting all vectors corresponding to dependent variables, constraint-handling method of penalty functions is applied to handle the inequality constraints related to dependent variables. Penalty factors corresponding to each dependent variable of each individual in population have to be calculated. If they violate a limit whether lower or upper, difference of that value and corresponding limit violated was taken as penalty index and it is multiplied with a constant so as to match with basic objective function i.e., fuel cost.

The penalty functions for slack bus power, voltages of load buses, line flows and reactive power generations are considered to calculate fitness of each population member. Fitness includes fuel cost function and also penalties corresponding to dependent variables. Inclusion of these penalties in fitness gives us a great opportunity to assign better fitness to that particular population member whose control parameters are within the operational limits in addition to minimum fuel cost

$$Fit = \cfrac{1}{\left[F_{cost} + K1 \cdot (P_{G1})^2 + K2 \cdot \sum_{i=1}^{Ng} Q_{gi}^2 \right.} \\ \left. + K3 \cdot \sum_{i=1}^{Ng} (V_{li})^2 + K4 \cdot \sum_{i=1}^{Ng} (F_{li})^2 \right]} \qquad (o)$$

where

Slack bus penalty → *Spf*

Line flow penalty → *Lfpf*

Q_G Penalty → *Qgpf*

Voltage penalty → *Vpf*

Differential Evolution Algorithm

1. Like any other evolutionary algorithm, DE also starts with a population of NP D-dimensional search variable vectors.

2. The subsequent generations in DE will be represented by discrete time steps like t = 0, 1, 2,..., t, t+1, etc.

3. Since the vectors are likely to be changed over different generations, the following notation may be adopted for representing the i th vector of the population at the current generation (i.e. at time t =t)as

Xi(t) =[xi,1(t), xi,2(t), ..., xi,D(t)]

These vectors are referred to as 'genomes' or 'chromosomes'.

4. Several optimization parameters must also be tuned. All needed parameters have joined together under the common name control parameters, although, as a matter of fact, there are only three real control parameters in the algorithm, which are:

(a) differentiation (or mutation) constant F,

(b) crossover constant CR, and (c) size of population NP.

Mathematical Formulation for Optimal Power Flow Model The mathematical formulation for OPF is based on the control variables and operating conditions (or) constraints.

Control Variables

(a) Generators active power outputs

(b) Generator bus voltages

(c) Controllable reactive compensation elements

(d) Transformer tap positions.

Constraints

Equality Constraints

The equality constraints are the active and reactive power balance equations at all the bus bars in each and every bus which are itself the load flow equations.

Inequality Constraints

The equality constraints are basically operating limits and physical limits of each equipment. That is active and reactive power limits, lines and transformers, transmission reactive power injection limits in the controlling tension bars and injection of active power in the reference bar.
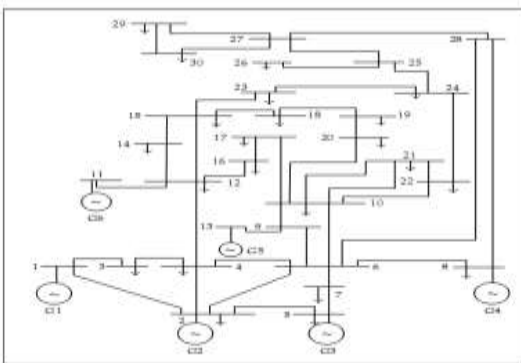

Figure 3: IEEE 30 bus test system

Model Formulation

Parameters

NG   □ Total number of generation including the slack bus.

i, j   □ Number of generated buses.

PG_i □ Generated active power output at bus i.

QG_i □ Generated reactive power output at bus i.

a_i, b_i, c_i □ Unit costs curve for i^{th} generator.

g_{ij} □ Conductance between buses i and j.

b_{ij} □ Susceptance between buses i and j.

V_i □ Voltage magnitude at the bust i.

δ_{ij} □ Voltage phase angle difference between i and j.

VG_i □ Generator voltage magnitude at the bus i.

VG_i^{min} □ Minimum generator voltage magnitude at the bust i.

VG_i^{max} □ Maximum generator voltage magnitude at the bust i.

PG_i^{min}, PG_i^{max} □ Lower and upper bounds of generated real power outputs at the bus i.

QG_i^{min}, QG_i^{max} □ Lower and upper bounds of generated reactive power outputs at the bus i.

NT   □ Total number of transformer tap settings.

T_i □ Number of transformer tap settings at bus i.

T_i^{min}, T_i^{max} □ Lower and upper bounds of transformer tap setting at the bus i.

NC   □ Total number of shunt VAR compensators.

QC_i □ Installation of reactive power for shunt VAR compensation at the bus i.

QC_i^{min}, QC_i^{max} □ Lower and upper bounds of shunt VAR compensation at the bus i.

NL   □ Total number of load buses.

PD_i □ Active power demand at bus i.

QD_i □ Reactive power demand at bus i.

$$\text{Min} \quad \sum_{i=1}^{NG} a_i + b_i PG_i + c_i PG_i^2$$

Subject to

$$\sum_{j=1}^{NG} \qquad \qquad \qquad \qquad \tag{1}$$

$$\sum_{j=1}^{NG} \qquad \qquad \qquad \qquad \tag{2}$$

$$V_i^{min} \leq V_i \leq V_i^{max} \qquad \tag{3}$$

$$PG_i^{min} \leq PG_i \leq PG_i^{max} \qquad \tag{4}$$

$$QG_i^{min} \leq QG_i \leq QG_i^{max} \qquad \tag{5}$$

$$T_i^{min} \leq T_i \leq T_i^{max} \qquad \tag{6}$$

$$QC_i^{min} \leq QC_i \leq QC_i^{max} \qquad \tag{7}$$

Solution Methodology

Lagrangian Relaxation Method

Relaxing Equations (1) and (2),

$$L = \qquad \qquad \qquad \qquad $$

$$\sum_{i=1}^{NG} \qquad \qquad \qquad \qquad $$

Subject to

$$V_i^{min} \leq V_i \leq V_i^{max} \qquad \tag{3}$$

$$PG_i^{min} \leq PG_i \leq PG_i^{max} \qquad \tag{4}$$

$$QG_i^{min} \leq QG_i \leq QG_i^{max} \qquad \tag{5}$$

$$T_i^{min} \leq T_i \leq T_i^{max} \qquad \tag{6}$$

$$QC_i^{min} \leq QC_i \leq QC_i^{max} \qquad \tag{7}$$

Lagrangian Relaxation replaces the original problem with an associated Lagrangian problem whose optimal solution provides a bound on the objective function of the problem. This is achieved by eliminating (relaxing one or more) constraints of

the original model and adding these constraints, multiplied by an associated Lagrangian multiplier in the objective function.

The main objective of this method is to relax the constraints that will result in a relaxed problem. When it gives the values of multipliers, it is much easier to solve optimally. The role of these multipliers is to derive the Lagrangian problem towards a solution that satisfies the relaxed constraints.

The Lagrangian relaxation approach replaces the problem of identifying the optimal values of all the decision variables with one of finding optimal or good values for the Lagrangian multipliers. Most Lagrangian-based heuristics use a search heuristic to identify the optimal multipliers. A major benefit of Lagrangian-based heuristics is that they generate bounds (i.e., lower bounds on minimization problems and upper bounds on maximization problems) on the value of the optimal solution of the original problem. For any set of values for the Lagrangian multipliers, the solution to the Lagangian model is less than or equal to the solution to the original model. Therefore, the Lagrangian solution is a lower bound on the solution to the original problem.

The solution to the Lagrangian problem for any given values of the Lagrangian multipliers will generally violate one or more of the relaxed constraints. Many Lagrangian based algorithms incorporate additional heuristics to convert these infeasible solutions to feasible ones. In this way, the researchers can produce good solutions to the original model. The best feasible solution among those found by the procedure at any point, represents the upper bound on the value of the true optimal solution. The difference between the upper and lower bounds is referred to as the "gap". If the gap reaches zero (or some minimum value based on the integer properties of the model) then the optimal solution should be found. Otherwise, when the gap gets sufficiently small (e.g. less than 1%), the analyst may stop the procedure and be satisfied that the current best solution is within 1% of optimality. The general application of Lagrangian relaxation can be found in Fisher (1985). An exposition of its use in location models is in the text by Daskin (1995). The proposed methodology has relaxed the power flow equations with respect to active power and reactive power. The Lagrangian function for OPF is minimize the total generation cost.

Conclusion

A Quadratic Programming (QP) model is designed for optimal power flow which can be decomposed by Lagrangian Relaxation Method. The solution of OPF is obtained to minimize the real power generation cost using Lagrangian relaxation method. The proposed methodology gives the appropriate power schedule for the operation of power system by minimizing the electrical losses in the system.

Acknowledgement

References

[1] J.B. Gupta, "A course in power systems", S.K. Kataria and Sons, Delhi.

[2] T.S. Chung and Ge Shaoyun, "A recursive L-p based approach for optimal capacitor allocation with cost-benefit consideration", Electric power system research, Vol. 39, pp. 129 137, 1997.

[3] E. Laboto, L. Rouco, M.I. Navarrete, R. Casanova and G. Lopez, "An LP-based optimal power flow for transimission losses and generator reactive margins minimization", in Proc. of IEEE Porto power tech conference, Portugal, Sep. 2001.

[4] S.D. Chen and J.F. Chen, "A new algorithm based on the Newton-Raphson approach for real-time emission dispatch", Electric power sys. research, Vol. 40, pp. 137 141, 1997.

[5] X. Tong and M. Lin, "Semismooth Newton-type algorithms for solving optimal power flow problems", in Proc. of IEEE/PES Transmission and distribution conference, Dalian, China, 2005, pp. 1 7.

[6] A. Momoh, "A generalized quadratic-based model for optimal power flow", CH 2809 2/89/0000 0261 $1.00© 1989 IEEE, pp. 261 267.

[7] N. Grudinin, "Reactive power optimization using successive quadratic programming method", IEEE Trans. Power syst., Vol. 13, No. 4, pp. 1219 1225, Nov. 1998.

[8] D. Pudjianto, S. Ahmed and G. Strbac, "Allocation of VAR support using LP and NLP based optimal power flows", IEEE Proc. Gener. Tranom. Distrib. Vol. 149, No. 4, pp. 377 383, July 2002.

[9] G.L. Torres and V.H. Quintana, "A jacobian smoothing non linear complementarity method for solving non linear optimal power flows", in proc. of 14th PSCC, Sevilla, Session 41, paper 1, June 2002, pp. 1 7.

[10] D. Xiaoying, W. Xifan, S. Yonghua and G. Jian, "The interior point branch and cut method for optimal power flow", 0 7803 7459 2/02/$17.00 © 2002 IEE, pp. 651 655.

[11] Wei Yan, J. Yu, D.C. Yu and K. Bhattarai, "A new optimal reactive power flow model in rectangular form and its solution by predictor corrector primal dual interior point method", IEEE Trans. Power Sys. Vol. 21, No. 1, pp. 61 67, Feb. 2006.

[12] N. Iwan Santoso and Owen T. Tan, "Neural-net based real time control of capacitors installed on distribution systems", IEEE Trans. Power delivery, Vol. 5, No. 1, pp. 266 272, Jan 1990.

[13] David C. Walters and Gerald B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loading", 92SM 414 3, IEEE/PES 1992. Summer Meeting.

[14] T.S. Chung and Y.Z. Li, "A hybrid GA approach for OPF with consideration of FACTS devices", IEEE power engineering review, pp. 47 50, Feb. 2001. H. Yoshida, K. Kawata, Y. Fukuyam et al., "A particle swarm optimization for reactive power and voltage control considering voltage security assessment", IEEE Trans power system, Vol. 15, No. 4, pp. 1232 1239, Nov. 2000.

[15] Cui-Ru Wang, He-Jinyuan, Zhi-Qian Huang, Jiang-Wei Zhang and Chen-Jun Sun, "A modified particle swarm optimization algorithm and its application in optimal power flow problem", in 4th international conference on machine learning and cybernetics, Guangzhon, August 2005, pp. 2885 2889.

[16] I.K. Yu and Y.H. Song, "A novel short-term generation scheduling technique of thermal units using ant colony search algorithms", Electrical power and energy sys. Vol. 23, pp. 471 479, 2001.

[17] P. Somasundaram, K. Kuppusamy and R.P.K. Devi, "Evolutionary programming based security constrained optimal power flow", Electric power sys. Research, 72, pp. 137 145, 2004.

S. Maheswari received B.Sc. Degree in Mathematics from the Madras University, Tamil Nadu, India, in 1997. She received M.Sc. Degree in Mathematics with Computer Applications from the Annamalai University, Tamil Nadu, India, in 1999. She received M. Phil. Degree in Mathematics from Alagappa University, Tamil Nadu, India, in 2006. She is currently working as a Assistant Professor in the Department of Mathematics in Sathyabama University, Tamil Nadu, India. Her area of interest is in the field of Non Linear Programming Problems and Optimization techniques.

Dr. C. Vijayalakshmi is currently working as Associate Professor in Mathematics Department, Sathyabama University, Chennai, Tamil Nadu. She has more than 15 years of teaching experience at Graduate and Post Graduate level. She has published more than fifteen research papers in International and National Journals and authored three books for Engineering colleges, Anna University. She has received Bharat Jyothi Award for outstanding services, achievements and contributions, Best Teachers award for academic excellence. Her area of specialization is Stochastic Processes and their applications. Her other research interests include Optimization Techniques, Data mining and Bio-Informatics.