MODERN OPTIMIZATION TECHNIQUES

TOPIC:

DESIGN OF A USER ORIENTED GENETIC ALGORITHM FOR MINIMIZATION OF LOSSES USING OPTIMAL POWER FLOW

SUBMITTED BY:

ARNAB BHATTACHARJEE

ROLL NO. 107115010

EEE-B

GUIDED BY:

MR. MUKESH MUTHU

SUBMITTED ON:

20/04/2018

ABSTRACT:

This project presents the design and implementation of a generic genetic algorithm function that provides multiple options to the user to choose from for optimizing its performance according to the need. Here we have used the algorithm to solve the optimal power flow problem for a n-bus system. Only continuous control variables, i.e., unit active power outputs and generator-bus voltage magnitudes were modeled. A number of functional operating constraints, such as slack bus real power limits, load bus voltage magnitude limits, and generator reactive capabilities, are included as penalties in the GA fitness function. Advanced and problem-specific operators are introduced in order to enhance the algorithm's efficiency and accuracy.

INTRODUCTION:

Since its introduction as network constrained economic dispatch by Carpentier and its definition as optimal power flow (OPF) by Dommel and Tinney, the OPF problem has been the subject of intensive research. The OPF optimizes a power system operating objective function (such as the operating cost of thermal resources) while satisfying a set of system operating constraints, including constraints dictated by the electric network. OPF has been widely used in power system operation and planning. After the electricity sector restructuring, OPF has been used to assess the spatial variation of electricity prices and as a congestion management and pricing tool. In its most general formulation, the OPF is a nonlinear, non-convex, large-scale, static optimization problem with both continuous and discrete control variables. Even in the absence of non-convex unit operating cost functions, unit prohibited operating zones, and discrete control variables, the OPF problem is non-convex due to the existence of the nonlinear (AC) power flow equality constraints. The presence of discrete control variables, such as switchable shunt devices, transformer tap positions, and phase shifters, further complicates the problem solution. Due to this reason we have restrained from using discrete variables in our solution. Mathematical programming approaches, such as nonlinear programming (NLP), quadratic programming (QP) and linear programming (LP) have been used for the solution of the OPF problem. OPF programs based on mathematical programming approaches are used daily to solve very large OPF problems. However, they are not guaranteed to converge to the global optimum of the general non-convex OPF problem, although there exists some empirical evidence on the uniqueness of the OPF solution within the domain of interest. The handling of non-convex OPF objective functions, as well as the unit prohibited operating zones, also present problems to mathematical programming OPF approaches.

Recent attempts to overcome the limitations of the mathematical programming approaches include the application of simulated annealing-type methods and genetic algorithms (GAs).

The GA-OPF approaches overcome the limitations of the conventional approaches in the modeling of non-convex cost functions, discrete control variables, and prohibited unit operating zones.

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. GAs 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 operate 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 use 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, unimodality, 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:

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.

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.

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.

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

REAL ENCODED GA:

The class of GA used in this project is the real encoded GA, in which we use real life values and not their binary equivalents. The use of real parameters makes it possible to use large domains (even unknown domains) for the variables, which is difficult to achieve in binary implementations where increasing the domain would mean sacrificing precision, assuming a fixed length for the chromosomes. Another advantage when using real parameters is their capacity to exploit the *graduality* of the functions with continuous variables, where the concept of graduality refers to the fact that slight changes in the variables correspond to slight changes in the function. In this line, a highlighted advantage of the RCGAs is the capacity for the *local tuning* of the solutions. There are genetic operators such as *non uniform mutation* (Michalewicz, 1992) that allows the tuning to be produced in a more suitable and faster way than in the BCGAs, where the tuning is difficult because of the Hamming cliff effect. Using real coding the representation of the solutions is very close to the natural formulation of many problems, e.g., there are no differences between the *genotype* (coding) and the *phenotype* (search space). Therefore, the coding and decoding processes that are needed in the BCGAs are avoided, thus increasing the GA's speed. In (Radcliffe, 1992) it is suggested that a distinction between genotype and phenotype is not necessary for evolution. Thus, it is not justified that the definition of the genetic operators should be made upon the representation chosen. On the contrary, the author argued that whenever possible, *genetic operators and the analogues of schemata should be directly defined in the space for phenotypes*. The genetic operators and the schema concepts presented in the literature for RCGAs agree with the Radcliffe's idea.

For Antonisse (Antonisse, 1989) binary coding is purposefully simple, while much of the work in the artificial intelligence community has been to develop highly expressive relatively complex representations. He presented a new schema interpretation that overturns the theoretical suitability of the binary alphabet in favour of the high cardinal alphabets. Further, for pointing out the importance of the expressiveness in the coding, he wrote:

": : : *This interpretation aligns theory with the intuition that the more expressive a language is the more powerful an apparatus for adaptation it provides, and encourages exploration of alternative encoding schemes in GA research.*"

Clearly, since with the use of real coding the genotype and phenotype are similar, the expressiveness level reached is very high. Along with these ideas, the relationship between the GAs and the *domain knowledge* need to be discussed. For Davis (Davis, 1989) most real world problems may not be handled using binary representations and an operator set consisting only of binary crossover and binary mutation. The reason is that nearly every real world domain has associated domain knowledge that is of use when one is considering a transformation of a solution in the domain. Davis believes that the real world knowledge should be incorporated into the GA, by adding it to the decoding process or expanding the operator set.

Real coding allows the domain knowledge to be easily integrated into the RCGAs for the case of problems with nontrivial restrictions, as we shall see below.

The selection techniques used in this are simple roulette wheel selection, tournament selection, random selection and stochastic universal sampling.

Roulette Wheel Selection:

Selection in this method is proportionate to the fitness of individual. Higher the fitness of individual, higher the chances of getting selected. The principle of roulette selection follows a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. The probability of an individual being selected as a parent for crossover is given by (Jebari & Madiafi, 2013),

$$p(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)}$$

Roulette Wheel Selection is the easiest and simplest method to implement and consumes the least amount of time. However it suffers from problem of premature convergence which results in finding a solution which is locally optimum.

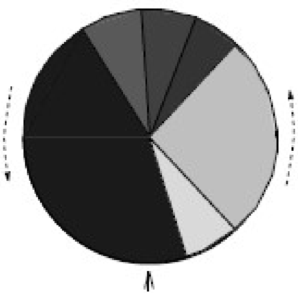
Tournament Selection:

In tournament selection, n individuals are selected from a large population. Then those n individual compete against each other. The one with the highest fitness wins and participates in crossover. Number of individuals competing against each other is termed as tournament size. Since tournament selection gives an equal chance to all the individuals to compete, hence, diversity is preserved. But this also leads to degradation in convergence speed (Razali & Geraghty, 2011). The probability of an individual being selected for reproduction is given by (Jebari & Madiafi, 2013),

$$p(i) = \begin{cases} \dfrac{C(k-1, n-1)}{C(k, n)} & if\ i \in [1, n-k-1] \\ 0\ if\ i\ \in [n-k, n] & \end{cases}$$

Stochastic Universal Sampling:

Stochastic universal sampling (SUS) is a technique used in genetic algorithms for selecting potentially useful solutions for recombination. It was introduced by James Baker. SUS is a development of fitness proportionate selection (FPS) which exhibits no bias and minimal spread. Where FPS chooses several solutions from the population by repeated random sampling, SUS uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals. This gives weaker members of the population (according to their fitness) a chance to be chosen and thus reduces the unfair nature of fitness-proportional selection methods.



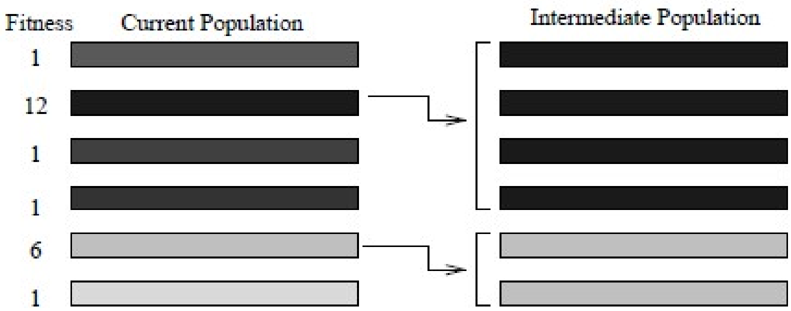


Figure 2. Selection application

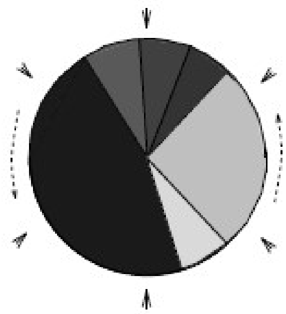


Figure 3. Stochastic universal sampling

Crossover Types:

Let us assume that C1=(c_1^1,…..,c_n^1) and C2=(c_1^2,…..,c_n^2) are two chromosomes that have been selected to apply the crossover operator to them. Below, the effects of different crossover operators for RCGAs are shown. We should point out that since each crossover operator generate a different offspring number, a selection mechanism for deciding the ones that shall be included in the population is sometimes needed. This selection mechanism shall be called *offspring selection mechanism*.

Simple crossover (Wright, 1991; Michalewicz, 1992)

A position $i \in \{1, 2, ..., n-1\}$ is randomly chosen and the two new chromosomes are built

$$H_1 = (c_1^1, c_2^1, ..., c_i^1, c_{i+1}^2, ..., c_n^2)$$

$$H_2 = (c_1^2, c_2^2, ..., c_i^2, c_{i+1}^1, ..., c_n^1)$$

Arithmetical crossover (Michalewicz, 1992)

Two offspring, $H_k = (h_1^k, ..., h_i^k, ..., h_n^k)$ $k = 1, 2$, are generated, where $h_i^1 = \lambda c_i^1 + (1-\lambda)c_i^2$ and $h_i^2 = \lambda c_i^2 + (1-\lambda)c_i^1$. λ is a constant (uniform arithmetical crossover) or varies with regard to the number of generations made (non-uniform arithmetical crossover).

BLX-α crossover (Eshelman et al., 1993)

An offspring is generated: $H = (h_1, ..., h_i, ..., h_n)$, where h_i is a randomly (uniformly) chosen number of the interval $[c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$, $c_{max} = \max(c_i^1, c_i^2)$, $c_{min} = \min(c_i^1, c_i^2)$, $I = c_{max} - c_{min}$. The BLX-0.0 crossover is equal to the flat crossover.

Wright's heuristic crossover (Wright, 1990)

Let's suppose that C_1 is the parent with the best fitness. Then $h_i = r \cdot (c_i^1 - c_i^2) + c_i^1$ and r is a random number belonging to $[0, 1]$.

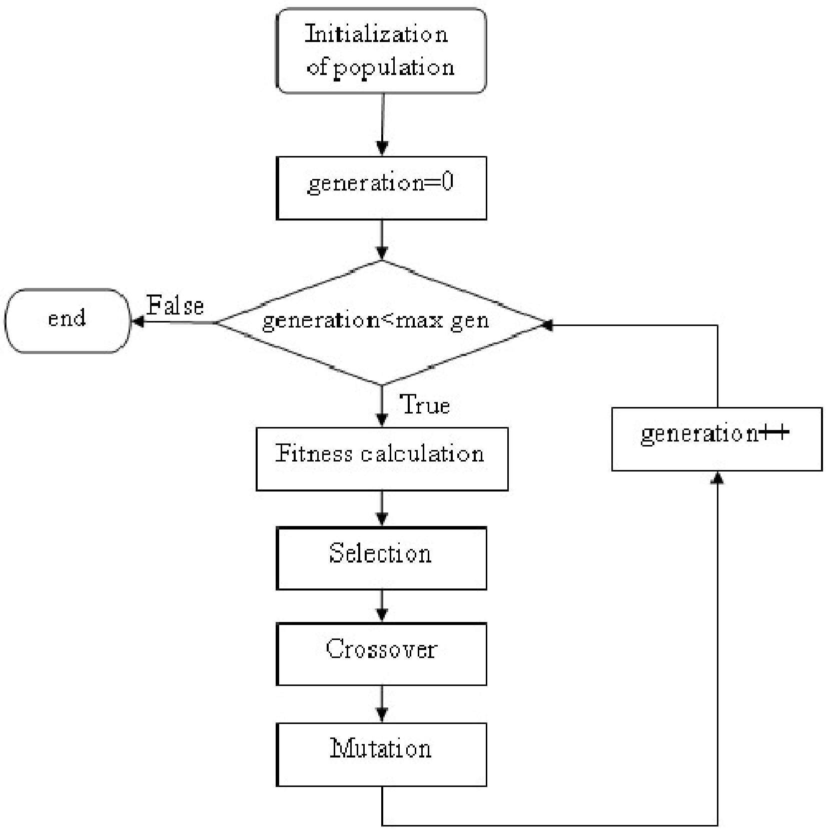
Linear BGA crossover (Schlierkamp-Voosen, 1994)

Under the same consideration as above, $h_i = c_i^1 \pm rang_i \cdot \gamma \cdot \Lambda$, where $\Lambda = \frac{c_i^2 - c_i^1}{||C_1 - C_2||}$.

The "$-$" sign is chosen with a probability of 0.9. Usually, $rang_i$ is $0.5 \cdot (b_i - a_i)$ and $\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$ where $\alpha_i \in \{0, 1\}$ is randomly generated with $p(\alpha_i = 1) = \frac{1}{16}$. This operator is based on *Mühlenbein's mutation* (Mühlenbein et al., 1993).

In the algorithm, the user is free to choose in between any of the selection methods and the crossover method to choose from and compare the results.

Flowchart for Algorithm:



OPTIMAL POWER FLOW PROBLEM FORMULATION:

The OPF problem can be formulated as a mathematical optimization problem as follows:

$$
\begin{aligned}
\text{Min} \quad & f(\mathbf{x}, \mathbf{u}) \\
\text{S.t.} \quad & g(\mathbf{x}, \mathbf{u}) = \mathbf{0} \\
& \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0} \\
& \mathbf{u} \in U
\end{aligned}
$$

- Independent or control variables (u)

This consists of the m operating variables that include active power of all generators except the slack generator and voltage magnitudes of all generators

$\mathbf{u}$=[$\mathbf{V_G}^{\mathbf{T}}$ $\mathbf{P_G}^{\mathbf{T}}$]

- Dependent or state variables (x)
  - This is the set of n = $2N_L - N_G - 2$ state variables.
  - Voltage angles of all buses excluding the slack bus and
  - voltage magnitudes at load buses

$\mathbf{x}$= [$\mathbf{V_L}^{\mathbf{T}}$ $\boldsymbol{\theta}^{\mathbf{T}}$ $\mathbf{P_{SG}}$ $\mathbf{Q_G}^{\mathbf{T}}$]

The equality constraints are the nonlinear power flow equations. The inequality constraints are the functional operating constraints, such as

• branch flow limits (MVA, MW or A);

• load bus voltage magnitude limits;

• generator reactive capabilities;

• slack bus active power output limits.

The fourth set of constraints define the feasibility region of the problem control variables such as

• unit active power output limits;

• generation bus voltage magnitude limits;

The optimal power flow objective that we will be using here is minimizing the active power losses in the system.

$$P_L = \Sigma P_i = \Sigma P_{gi} - \Sigma P_{di;}, i = 1, \ldots\ldots\ldots\ldots\ldots, N_b$$

PERFORMANCE FOR STANDARD PROBLEMS:

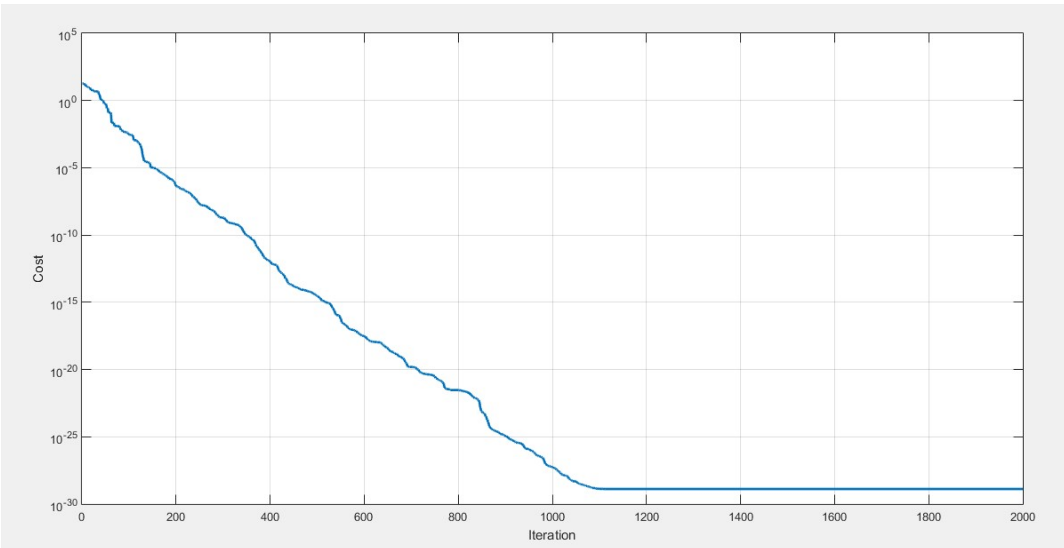
- LEVY FUNCTION
  TYPE OF CROSSOVER: Wright's Heuristic Crossover

  TYPE OF SELECTION: Tournament Selection

```
What selection method you want to employ: 1. Roulette Wheel Selection  2. Tournament Selection  3. Random Selection  4. Stochastic Universal Sampling?  2
What crossover type do you want : 1. Simple crossover  2.Arithmetical Crossover  3.BLX-alpha  4. Wrights Heuristic  5. Linear BGA ?  4
Best Solution is :
    Position: [1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1]
        Cost: 1.3018e-29
```

Best Cost = 1.3018 * 10 ^ (-29)

Best Solution = [1, 1, 1, 1, 1, 1, 1]



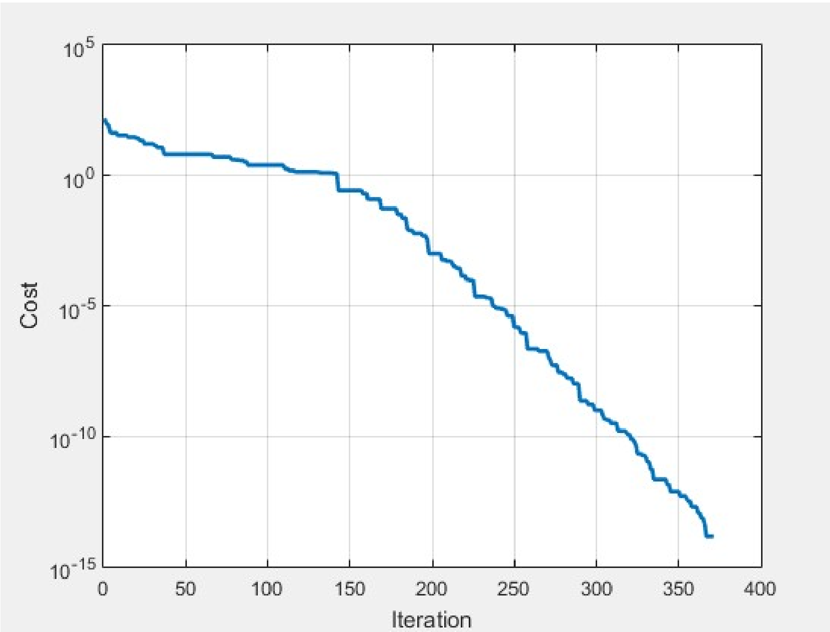
- RASTRIGIN FUNCTION
  TYPE OF CROSSOVER: BLX-alpha Crossover

  TYPE OF SELECTION: Roulette Wheel Selection

```
What selection method you want to employ: 1. Roulette Wheel Selection  2. Tournament Selection  3. Random Selection  4. Stochastic Universal Sampling?  1
What crossover type do you want : 1. Simple crossover  2.Arithmetical Crossover  3.BLX-alpha  4. Wrights Heuristic  5. Linear BGA ?  3
Best Solution is :
    Position: [-9.2206e-10 4.8432e-09 -1.5369e-09 -1.4979e-09 1.7999e-09 2.9392e-09 4.6124e-09]
        Cost: 0
```

Best Cost= [0,0,0,0,0,0,0]

Best Cost= 0



OPF WITH GENETIC ALGORITHM:

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.

2) 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).

A. Encoding

There are two chromosome regions (one for each set of control variables), namely, 1) P_G 2)V_G. Real coded genetic algorithm is used. The decoding of a chromosome to the problem physical variables is performed as follows: 1) continuous controls taking values in the interval

$$[u_i^{\min}, u_i^{\max}]$$

$$u_i = u_i^{\min} + (u_i^{\max} - u_i^{\min}) \cdot \frac{k}{2^{N_{u_i}} - 1}$$

B. Fitness Function (FF)

The objective of the OPF problem is to minimize the total active power loss. Therefore, a transformation is needed to convert the cost objective of the OPF problem to an appropriate FF to be minimized 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 are automatically satisfied by the selected GA encoding/decoding scheme. Therefore, the GA FF is formed as follows:

$FF= P_L + \sum_{j=1 \text{ to } Nc} Pen_j$

$$P_L = \Sigma P_i = \Sigma P_{gi} - \Sigma P_{di;}, i = 1, \ldots\ldots\ldots\ldots\ldots, N_b$$

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

where

FF	fitness function;
Pen_j	penalty function for functional operating constraint j;
$h_j(\mathbf{x}, \mathbf{u})$	violation of jth functional operating constraint, if positive;
$H(\cdot)$	Heaviside (step) function;

Nc = Number of operating constraints

N_b= Number of buses

Algorithm:

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, **u** .The computed control vector satisfies, by design, all the given constraints.

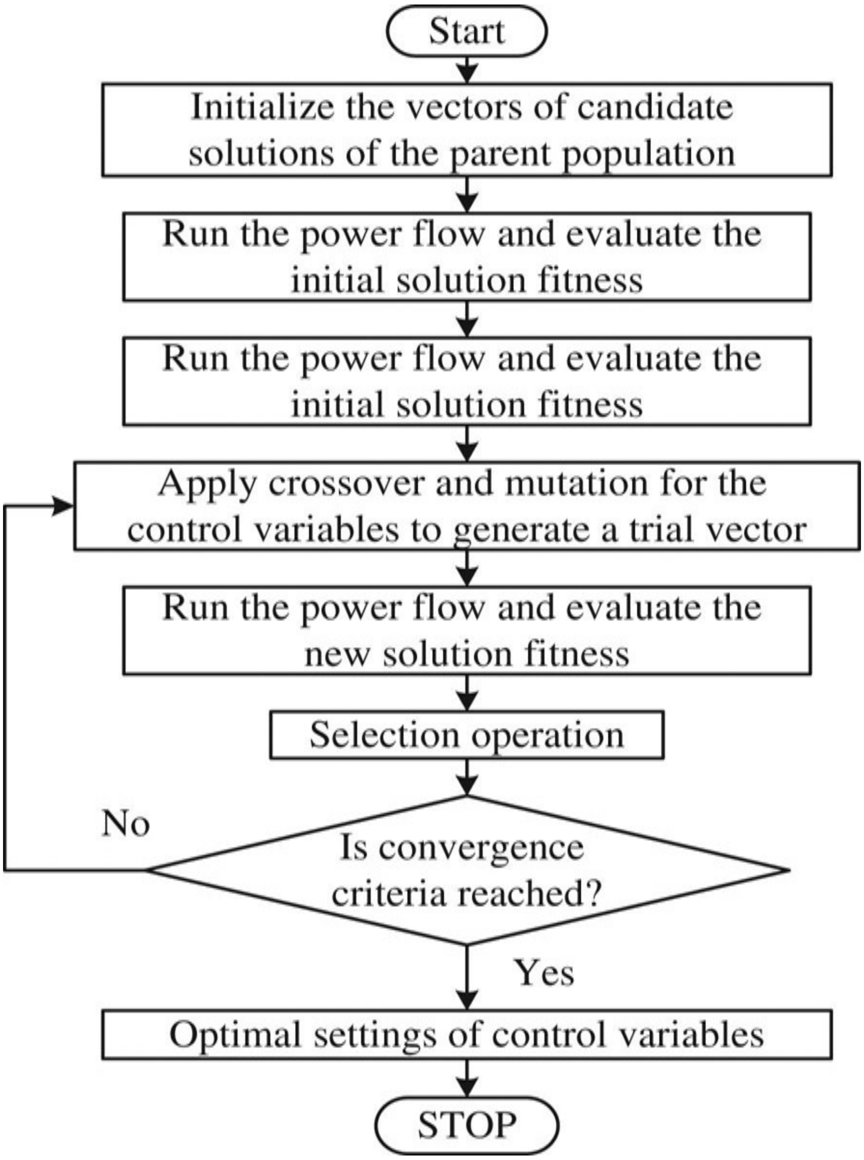
Step 2) Solve the power flow to compute the state vector, **x**.

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

Step 4) Compute the FF

In Step 2, a simple newton-raphson based load flow (NRLF) is used with no PV-PQ bus-type switching, since generator reactive capabilities are incorporated in the functional operating constraints. Therefore, only a few load flow iterations are required for convergence. The NRLF and matrices are formed and factorized only once in the beginning—the effect of the changes of shunt admittances on the matrix is neglected. A 30 bus system was used and the load flow was carried out using standard MATPOWER 6.0 functions.

Flowchart for specialized problem:

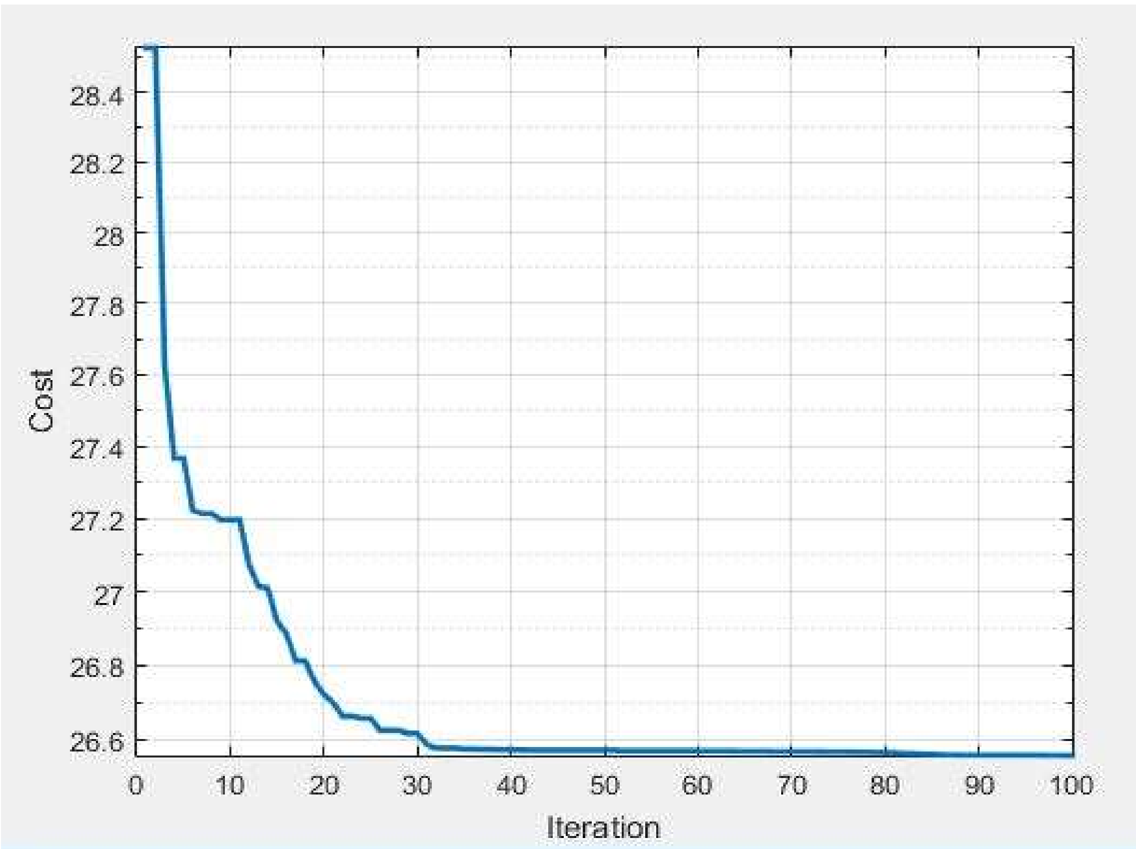


GA PERFORMANCE ANALYSIS FOR CHOSEN PROBLEM:

1. Selection: Tournament
   Crossover: Wright's Heuristic

   Loss = 1.698 MW

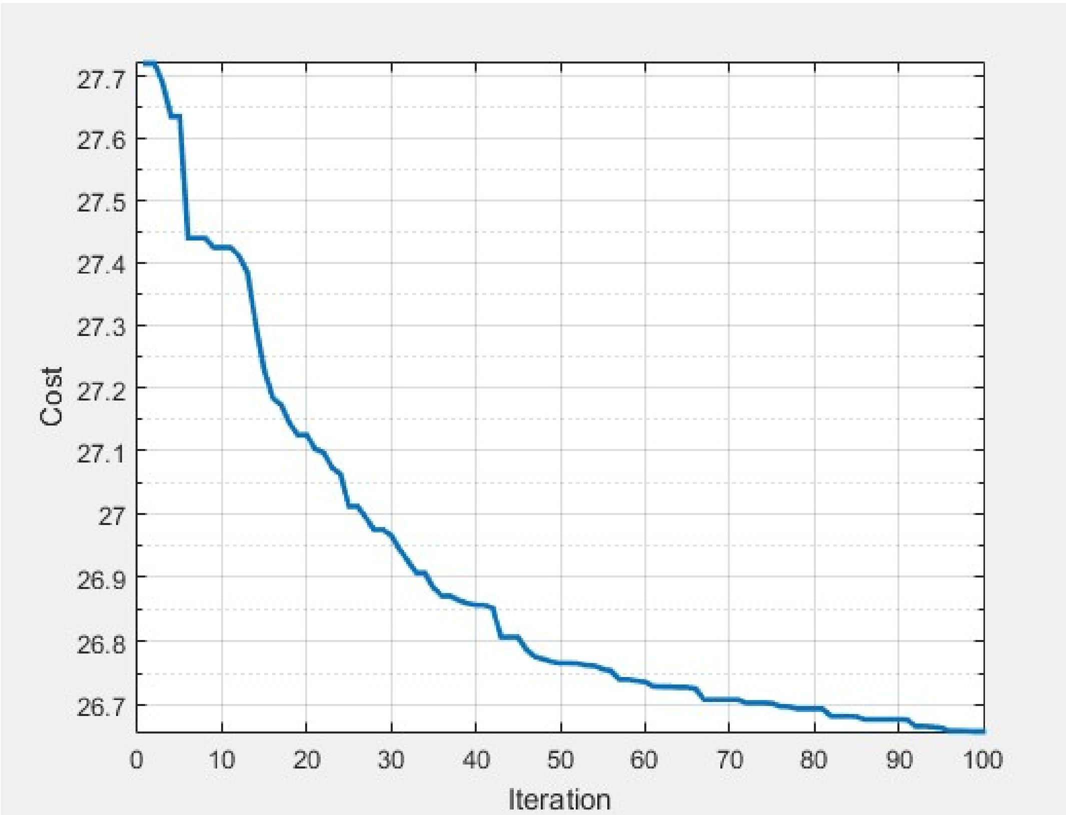
```
                                                           --------   --------
                                                  Total:     1.698      7.34
Iteration 100: Best Cost = 26.5565
Best Solution is :
    Position: [48.2637 50 30.5830 16.2121 40 1.0504 1.0499 1.0556 1.0600 1.0537 1.0600]
        Cost: 26.5565
```



2. Selection = Roulette Wheel
   Crossover= Wright's Heuristic

   Loss= 1.769 MW

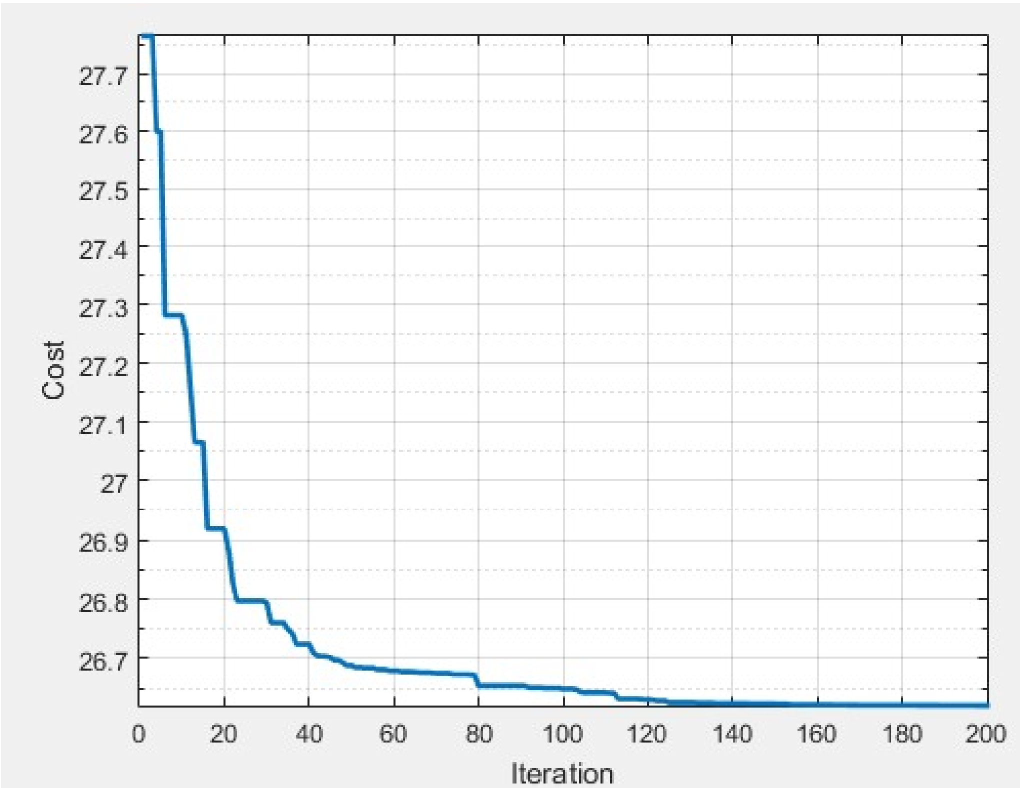
```
                                                        --------  --------
                                          Total:         1.769      9.33
Iteration 100: Best Cost = 26.6589
Best Solution is :
    Position: [46.3051 50 32.7099 16.3610 40 0.9892 0.9884 1.0164 1.0236 1.0161 1.0600]
        Cost: 26.6589
```



3. Selection = Stochastic Universal Sampling
   Crossover= BLX-alpha

   Loss= 1.730 MW

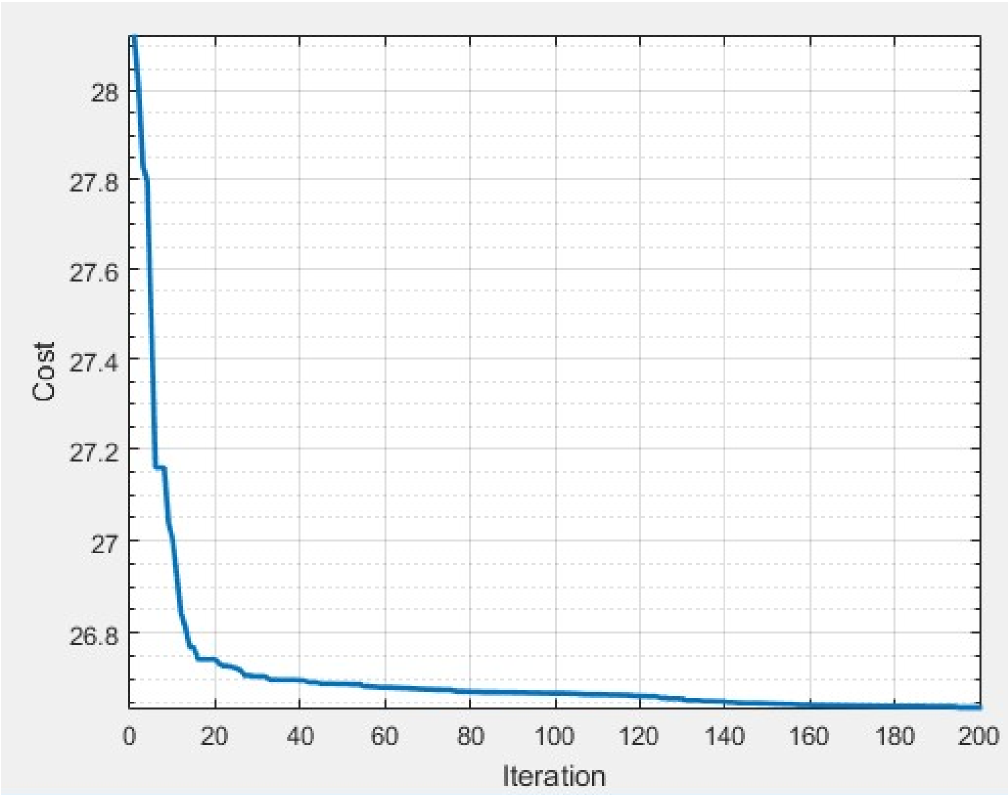
```
                                          Total:      1.730       9.13
Iteration 200: Best Cost = 26.6215
Best Solution is :
    Position: [46.9524 49.9995 32.0483 15.7879 40 0.9968 0.9981 1.0237 1.0347 1.0256 1.0600]
        Cost: 26.6215
```



4. Selection = Random Selection
   Crossover= Arithmetic Crossover

   Loss= 1.762 MW

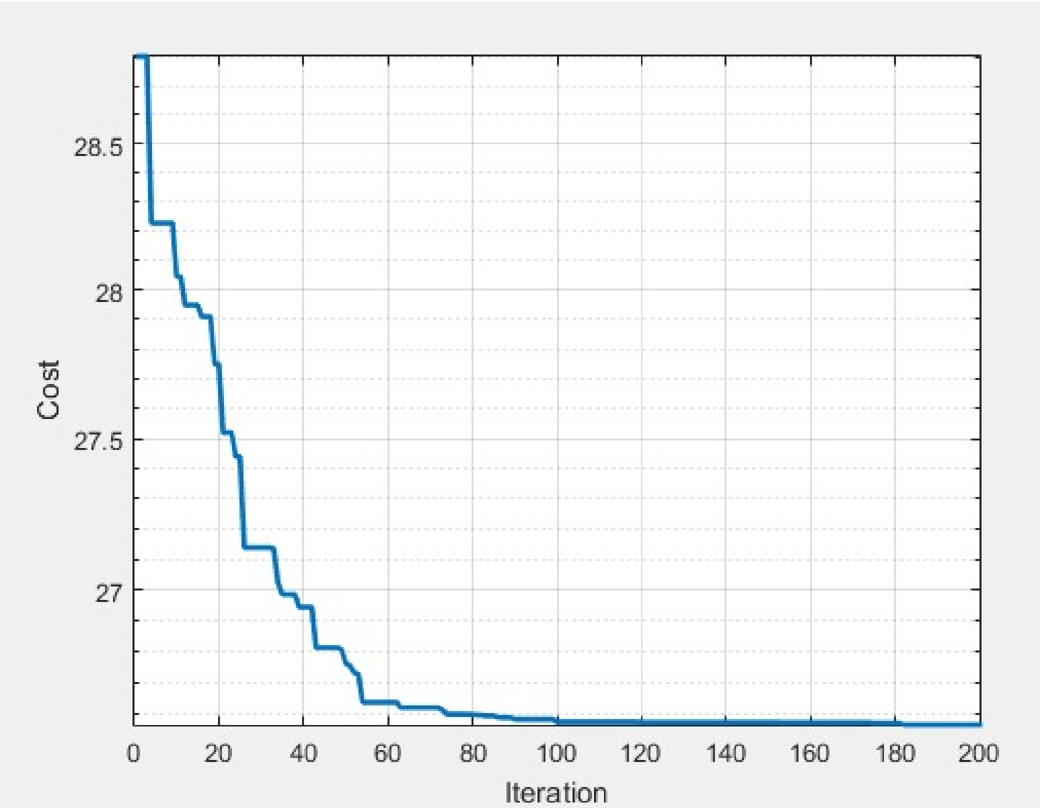
```
                                                         Total:     1.762      9.47
Iteration 200: Best Cost = 26.638
Best Solution is :
    Position: [47.3750 49.9570 31.0977 15.6634 39.9994 0.9933 0.9942 1.0164 1.0279 1.0206 1.0588]
        Cost: 26.6380
```



5. Selection = Random
   Crossover = Linear BGA

   Loss= 1.703 MW

```
                                                        Total:     1.703      6.91
Iteration 200: Best Cost = 26.5628
Best Solution is :
    Position: [48.3054 50 29.6193 15.5113 40 1.0519 1.0489 1.0566 1.0600 1.0562 1.0600]
        Cost: 26.5628
```



FINAL LOAD FLOW DATA:

Here selection used is Tournament Selection and crossover is Wright's Heuristic. This is the final load flow data obtained.

```
Newton's method power flow converged in 3 iterations.

Converged in 0.01 seconds
================================================================================
|     System Summary                                                           |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses              30    Total Gen Capacity     335.0           -95.0 to 405.9
Generators          6    On-line Capacity       335.0           -95.0 to 405.9
Committed Gens      6    Generation (actual)    190.9              97.8
Loads              20    Load                   189.2             107.2
  Fixed            20      Fixed                189.2             107.2
  Dispatchable      0      Dispatchable          -0.0 of -0.0      -0.0
Shunts              2    Shunt (inj)             -0.0               0.2
Branches           41    Losses (I^2 * Z)         1.67              7.78
Transformers        0    Branch Charging (inj)     -               16.9
Inter-ties          7    Total Inter-tie Flow    45.5              28.8
Areas               3

                          Minimum                      Maximum
                 -------------------------  --------------------------------
Voltage Magnitude   1.009 p.u. @ bus 8          1.060 p.u. @ bus 13
Voltage Angle      -1.36 deg   @ bus 7          3.55 deg   @ bus 13
P Losses (I^2*R)             -                  0.15 MW    @ line 27-30
Q Losses (I^2*X)             -                  2.42 MVAr  @ line 12-13
```

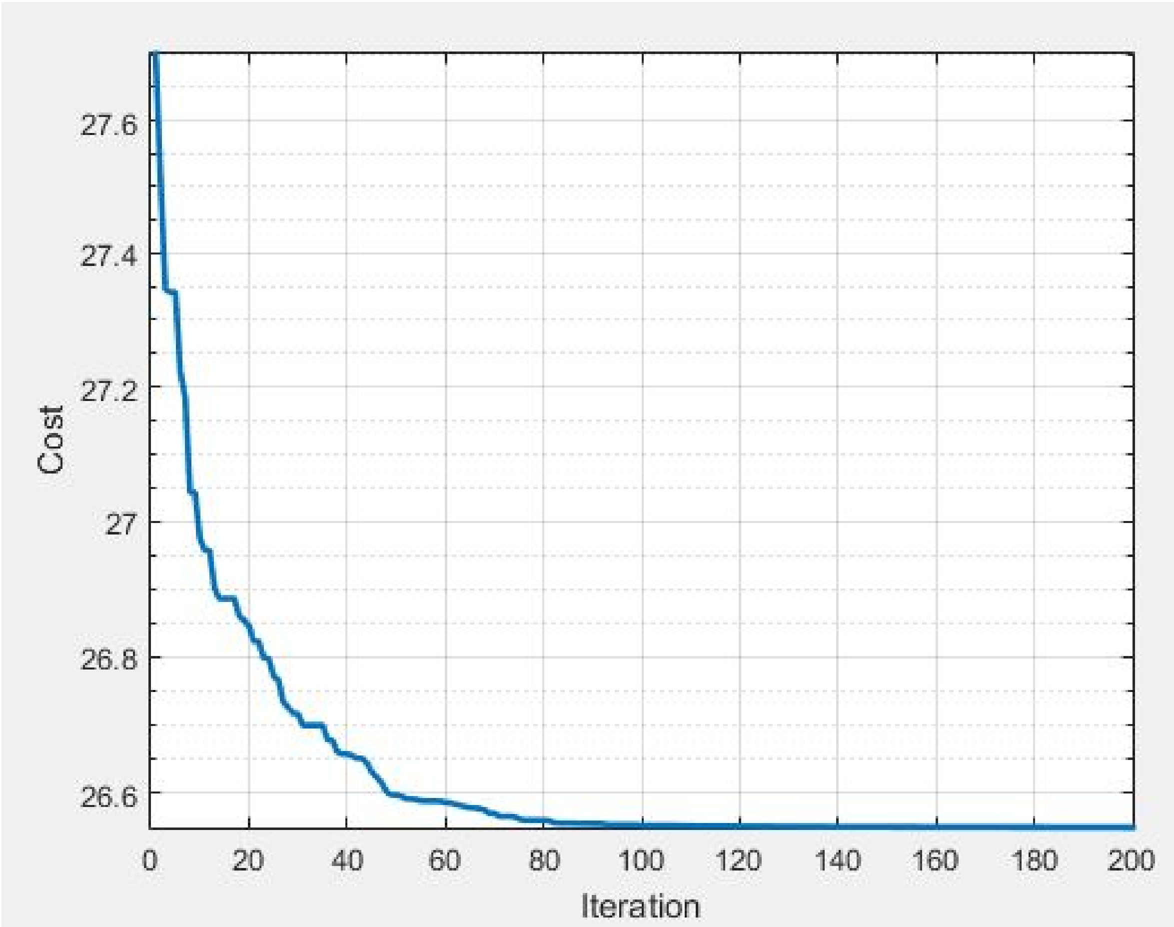
Bus Data

Bus #	Voltage Mag(pu)	Voltage Ang(deg)	Generation P (MW)	Generation Q (MVAr)	Load P (MW)	Load Q (MVAr)
1	1.038	0.000*	7.57	1.12	-	-
2	1.038	-0.041	47.36	22.90	21.70	12.70
3	1.028	-0.537	-	-	2.40	1.20
4	1.027	-0.601	-	-	7.60	1.60
5	1.025	-0.915	-	-	-	-
6	1.021	-0.817	-	-	-	-
7	1.014	-1.357	-	-	22.80	10.90
8	1.009	-1.207	-	-	30.00	30.00
9	1.027	-0.221	-	-	-	-
10	1.030	0.089	-	-	5.80	2.00
11	1.027	-0.221	-	-	-	-
12	1.037	0.634	-	-	11.20	7.50
13	1.060	3.555	40.00	18.54	-	-
14	1.028	-0.017	-	-	6.20	1.60
15	1.030	0.046	-	-	8.20	2.50
16	1.027	0.113	-	-	3.50	1.80
17	1.024	-0.112	-	-	9.00	5.80
18	1.018	-0.653	-	-	3.20	0.90
19	1.014	-0.879	-	-	9.50	3.40
20	1.017	-0.689	-	-	2.20	0.70
21	1.040	0.571	-	-	17.50	11.20
22	1.047	0.825	50.00	29.69	-	-
23	1.047	0.591	11.68	9.80	3.20	1.60
24	1.038	0.688	-	-	8.70	6.70
25	1.046	1.406	-	-	-	-
26	1.029	1.004	-	-	3.50	2.30
27	1.060	2.105	34.27	15.75	-	-
28	1.025	-0.657	-	-	-	-
29	1.041	0.951	-	-	2.40	0.90
30	1.030	0.143	-	-	10.60	1.90
		Total:	190.87	97.80	189.20	107.20

Brnch #	From Bus	To Bus	From Bus Injection P (MW)	From Bus Injection Q (MVAr)	To Bus Injection P (MW)	To Bus Injection Q (MVAr)	Loss (I^2 * Z) P (MW)	Loss (I^2 * Z) Q (MVAr)
1	1	2	1.31	-1.58	-1.30	-1.66	0.000	0.00
2	1	3	6.26	2.70	-6.24	-4.74	0.025	0.09
3	2	4	7.63	3.19	-7.59	-5.20	0.043	0.12
4	3	4	3.84	3.54	-3.84	-3.53	0.003	0.01
5	2	5	9.21	3.31	-9.16	-5.24	0.048	0.19
6	2	6	10.12	5.35	-10.04	-7.23	0.080	0.24
7	4	6	12.65	11.10	-12.63	-10.99	0.027	0.11
8	5	7	9.16	5.44	-9.11	-6.35	0.057	0.14
9	6	7	13.75	3.68	-13.69	-4.55	0.060	0.16
10	6	8	23.42	23.68	-23.32	-23.26	0.106	0.43
11	6	9	-5.20	-2.71	5.20	2.78	0.000	0.07
12	6	10	-2.97	-1.55	2.97	1.61	0.000	0.06
13	9	11	0.00	0.00	0.00	0.00	0.000	0.00
14	9	10	-5.20	-2.78	5.20	2.81	0.000	0.04
15	4	12	-8.83	-3.97	8.83	4.20	0.000	0.23
16	12	13	-40.00	-16.12	40.00	18.54	0.000	2.42
17	12	14	5.26	1.32	-5.23	-1.25	0.033	0.07
18	12	15	8.73	0.56	-8.68	-0.47	0.050	0.09
19	12	16	5.98	2.54	-5.95	-2.46	0.035	0.08
20	14	15	-0.97	-0.35	0.98	0.36	0.002	0.00
21	16	17	2.45	0.66	-2.44	-0.65	0.005	0.01
22	15	18	7.03	2.41	-6.97	-2.30	0.057	0.11
23	18	19	3.77	1.40	-3.76	-1.38	0.009	0.02
24	19	20	-5.74	-2.02	5.75	2.05	0.011	0.03
25	10	20	8.01	2.89	-7.95	-2.75	0.062	0.14
26	10	17	6.58	5.20	-6.56	-5.15	0.020	0.05
27	10	21	-16.40	-8.25	16.49	8.47	0.095	0.22
28	10	22	-12.16	-6.27	12.28	6.53	0.124	0.26
29	21	22	-33.99	-19.67	34.14	19.95	0.143	0.29
30	15	23	-7.52	-4.80	7.60	4.95	0.075	0.15
31	22	24	3.58	3.21	-3.56	-3.17	0.025	0.04
32	23	24	0.88	3.25	-0.87	-3.22	0.013	0.03
33	24	25	-4.28	-0.27	4.31	0.32	0.032	0.06
34	25	26	3.54	2.36	-3.50	-2.30	0.041	0.06
35	25	27	-7.85	-2.69	7.92	2.82	0.069	0.13
36	28	27	-13.08	-8.69	13.08	9.63	0.000	0.94
37	27	29	6.16	1.65	-6.08	-1.50	0.080	0.15
38	27	30	7.10	1.64	-6.95	-1.35	0.151	0.28
39	29	30	3.68	0.60	-3.65	-0.55	0.031	0.06
40	8	28	-6.68	-6.74	6.73	4.83	0.046	0.15
41	6	28	-6.35	-4.88	6.36	3.87	0.011	0.03
						Total:	1.669	7.78

```
Iteration 200: Best Cost = 26.5506
Time elapsed in the optimiation is
  144.2706

Best Solution is :
    Position: [47.4258 50 31.1392 15.7295 40 1.0383 1.0381 1.0474 1.0600 1.0471 1.0600]
        Cost: 26.5506
```


Cost
27.6
27.4
27.2
27
26.8
26.6
0
20
40
60
80
100
120
140
160
180
200
Iteration

CONCLUSION:

A GA solution to the OPF problem has been presented and applied to a medium size power system. The main advantage of the GA solution to the OPF problem is its modeling flexibility: non-convex unit cost functions, prohibited unit operating zones, discrete control variables, and complex, nonlinear constraints can be easily modeled. Another advantage is that it can be easily coded to work on parallel computers. The main disadvantage of GAs is that they are stochastic algorithms and the solution they provide to the OPF problem is not guaranteed to be the optimum. Another disadvantage is that the execution time and the quality of the provided solution deteriorate with the increase of the chromosome length, i.e., the OPF problem size. The applicability of the GA solution to large-scale OPF problems of systems with several thousands of nodes, utilizing the strength of parallel computers, has yet to be demonstrated.

REFERENCES:

- **A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas and V. Petridis, "Optimal power flow by enhanced genetic algorithm," in *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 229-236, May 2002. doi: 10.1109/TPWRS.2002.1007886 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1007886&isnumber=21737**
- **Herrera, F., Lozano, M. , Verdegay, J.L. PY - 1998, DA - 1998/08/01, TI - Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, JO - Artificial Intelligence Review, SP - 265, EP - 319, VL - 12, IS - 4, AB**
- **Class notes**