A Hybrid Genetic Algorithm Approach for Optimal Power Flow

Mithun M. Bhaskar*, Sydulu Maheswarapu
Dept of Electrical Eng., National Institute of Technology Warangal, Andhra Pradesh, India
e-mail: mithun.bm@ieee.org, mmbaskr@gmail.com

Abstrak

Makalah ini mengajukan sebuah pendekatan aliran daya optimal berbasis algoritma genetik (GA) hibrida yang dibentuk ulang. Pada pendekatan ini, variabel kontinyu dirancang menggunakan GA tersandi real dan variabel diskrit diproses sebagai string biner. Hasil ini dibandingkan dengan metode lain seperti algoritma genetik sederhana (GA), algoritma genetik adaptif (AGA), evolusi diferensial (DE), optimasi partikel swarm (PSO) dan pencarian harmoni berbasis music (MBHS) pada tes bus IEEE30, dengan beban total 283,4 MW. Hasil penelitian menunjukkan bahwa algoritma yang diusulkan menawarkan biaya bahan bakar terendah. Metode yang diusulkan memiliki komputasi lebih cepat, kuat, unggul dan menjanjikan bentuk karakteristik konvergensi.

Kata kunci: *algoritma genetik hibrida, aliran daya optimal, IEEE30 bus, teknik optimasi*

Abstract

This paper puts forward a reformed hybrid genetic algorithm (GA) based approach to the optimal power flow. In the approach followed here, continuous variables are designed using real-coded GA and discrete variables are processed as binary strings. The outcomes are compared with many other methods like simple genetic algorithm (GA), adaptive genetic algorithm (AGA), differential evolution (DE), particle swarm optimization (PSO) and music based harmony search (MBHS) on a IEEE30 bus test bed, with a total load of 283.4 MW. It's found that the proposed algorithm is found to offer lowest fuel cost. The proposed method is found to be computationally faster, robust, superior and promising form its convergence characteristics.

Keywords: *hybrid genetic algorithm, IEEE30 bus, optimal power flow, optimization techniques*

1. Introduction

The rapid growth in power system structures marked with open-access has brought in qualmswith worries on the possibility and impact of a ruminant security outage with the recent blackouts around the world. Increasing demand and liberalized option lead the system operators to work on narrow spinning reserve and to operate on vicinities to capitalize the economy compromising on the reliability and security of the system for greater profits. This lead to the inevitability of a monitoring authority and accurate electronic system to prevent any untoward incidents and to optimize the system controls for a greater economy.

A modern energy management system (EMS) caters to the ever increasing demands where the consumer depends not only on the availability of the electricity, but also looks for reliable, secure, superior and uninterrupted supply. Optimal power flow (OPF) problem is the perfect incorporation of the contradictory doctrines of maximum economy, safer operation and augmented security. OPF refers to the generator dispatch and resulting AC power flows at minimum and feasible cost with respect to thermal limits on the AC transmission lines. The OPF might include other constraints such as interface limits and other decisions such as the optimal flow on DC lines and phase shifter angles. OPF is a static nonlinear programming problem which optimizes a certain objective function while satisfying a set of physical and operational constraints imposed by equipment limitations and security requirements. In general, OPF problem is a large dimension nonlinear, non-convex and highly constrained optimization problem.

The first approach to OPF has been made by J Carpentair in 1962 and much of the significant developments from then are reported in [1-2]. Classical methods like Newton method,

Received February 14th, 2011; Revised May 2nd, 2011; Accepted May 9th, 2011

Gradient, Linear and Non-Linear programming, Quadratic programming etc., which suffered from slow convergence and some sort of inferiority which leads to the implantation of Artificial Intelligence and Evolutionary programming methods [3]. Meta-heuristics like Animal flocking, Ant colony, Tabu Search etc., emerged to be another alternative and a comparison of meta-heuristic methods for many trial systems can be found in [4].

Genetic Algorithm has been fascinating the researchers for long and first credits owe to John Holland in 1970. Genetic Algorithm, a stochastic routine [5], delivers quality solution from a random search space and population, where each generation undergoes transformation using Genetic operators to improve them. Genetic Algorithm transvers multiple peaks in parallel, assuring global solutions and has the advantage of modeling discrete [6] and continuous variables together which is not available with other algorithms. Much variants and developments in Genetic Algorithm has been discussed in [7]. As OPF being multi-model in nature, all these methods had the shortcoming of settling in a local minimum than a global optimal solution and being a much of approximation and dependent on continuous variables, designing with discrete variable became a concern. Here, the GA proved to provide a superior solution and much works has been reported using the conventional GA and its variants. Here, a hybrid GA is proposed which inherits the superiority of the conventional binary GA and the real coded GA.

2. Hybrid Genetic Algorithm

GA works with a population of solutions instead of a single solution. As there is more than one string being processed simultaneously and used to update every string in the population, it is likely that the expected GA solution may be global solution. GA work directly with a coding of decision variables, instead of the variables themselves. They work with a discrete search space, even though the function may be continuous.When binary coded GAs need to be used to handle problems having a continuous search space, a number of difficulties arise [8] including converging in near-optimal solutions and consumption of higher computational power. Other difficulties include, hamming cliffs associated with certain strings (such as strings 01111 and 10000) from which a transition to a neighboring solution (in real space) requires the alteration of many bits. Hamming cliffs present in a binary coding cause artificial hindrance to a gradual search in the continuous search space. The other difficulty is the inability to achieve any arbitrary precision in the optimal solution. Hence a Hybrid; binary and real coded genetic algorithm is presented here.

Binary coded GAs inherit the advantage that decision variables are coded in finite length strings and exchanging portions of two parent strings easier to implement and visualize. Whereas Real Coded GA has the advantage that the real parameters can be used intact and crossover and mutation operators are applied directly to real parameter values. Since the selection operator works with the fitness value, any selection operator used with binary coded GAs can also be used in real parameter GAs. But, In real parameter GAs, the main challenge is how to use a pair of real parameter decision variable vectors to a mutated vector in a meaningful manner as in such cases the term „crossover" is not that meaningful, they can be best described as blending operators.

2.1 Constraint Handling

The nonexistence of a feasible solution, it's essentially means that too many constraints have been added to the problem and no solution exists which obeys all of the constraints. One way to avoid this issue is to implement soft inequality constraints in the form of penalty functions. Here, in the OPF problem, use of soft-constraints are limited as in equality constraints, the power flow equations cannot be violated as they are imposed by physics, and the generator set points of a Power System are normally not moved around frequently. For the inequality constraints, the penalty functions offer a viable option.So, penalty functions are added to the objective function of the OPF. Ideally, a penalty function will be very small, near a limit and increase rapidly as the limit is violated more. The penalty function is zero when the inequality constraint are not violated and as the constraint begins to be violated, the penalty function quickly increases and reduces on reduction in violation limits.

2.2 Representation of an Individual

The individual chromosome s is composed of the continuous control variables x and the discrete control variables u, i.e., $s = [x, u]$. Each control variable is a gene. The encoding the physical variable is performed as follows. Continuous variables xi taking the real value in the interval, $[x_i^{min}, x_i^{max}]$ and discrete variable discrete variable $u_i$taking the decimal integer value$n_i$in the interval$[0 \dots \dots \mathrm{M}_i]$, where

$$\mathrm{M}_i = int\left(\left(u_i^{max} - u_i^{min}\right)/ST_i\right) \quad (1)$$

$$u_i = u_i^{min} + n_i * \ ST_i \quad (2)$$

STi : adjustable step size of the discrete control variable ui
int ( ) : the operator of rounding the variable to a nearest integer.

According to the above encoding schemes, the representation of the jth individual (t) in the population could be put forward as,

$$S_j(t) = \left[P_{G2}, \dots . P_{Gn}, V_1, \dots . V_n, T_{p1}, \dots . T_{pN}, Y_{h1}, \dots . Y_{hN}\right]_{\mathrm{j}} \quad (3)$$

where, j=1, 2… pz, (pz is population size) and N is the number of control variables.

2.3 Genetic Operators

The fitness of OPF, being a minimization problem, the fitness is calculated as the reciprocal of the objective. The primary objective of Genetic Operators is to make duplicates of the best fit solutions in the population and eliminate least fit solutions, while keeping the population size constant.Most commonly used method is Roulette Wheel Selection (RWS). Where the wheel is divided into N (population size) divisions and the size of each is marked in proportion to the fitness of each population member.

Elitism scheme ensures that the best found solution found, so far is never lost when moving from one generation to another generation, herein 15% elitism is considered.

2.4 Cross over Technique

Crossover recombines together good substrings from two good strings to hopefully from a better substrings and two arithmetic crossover schemes are employed [9]to treat the continuous and discrete variables and to produce two types of substrings, respectivelyand a cross over probability of (P_C) of 90% is used.

$PG$and V are continuous variables in string S(t), let the two individuals $S_1(t)$ and $S_2(t)$be crossed to produce two substrings $S_3(t)$ and $S_4(t)$as a linear combination of their parents (1) and S(2), i.e.,

$$\mathrm{S}_3(\mathrm{t}) = \lambda * \mathrm{S}_1(\mathrm{t}) + (1 - \lambda) * \mathrm{S}_2(\mathrm{t}) \quad (4)$$

$$S_4(t) = (1 - \lambda) * S_1(t) + (\lambda) S_2(t) \quad (5)$$

where astap setting T_P and switchable shunt device Y_h are the discrete variables in string S(t). Then, the two individuals $S_5(t)$and$S_6(t)$are to be crossed to produce two substrings $S_7(t)$and $S_8(t)$as the linear combination of their parents $S_5(t)$ and$S_6(t)$.

$$\mathrm{S}_7(\mathrm{t}) = \mathrm{int}[\ \lambda * \mathrm{S}_5(\mathrm{t}) + (1 - \lambda) * \mathrm{S}_6(\mathrm{t})] \quad (6)$$

$$\mathrm{S}_8(\mathrm{t}) = \mathrm{int}\ [1 - \lambda) * \mathrm{S}_5(\mathrm{t}) + (\lambda)\mathrm{S}_6(\mathrm{t})] \quad (7)$$

where,$S_6(t)$, $S_7(t)$,$S_8(t)$ ϵ $\{T\ , h\}$, λ is a uniform random positive number in the range of 0 to 1. The crossover is applied with a probability PC in the range of 0.8 to 0.9.

2.5 Mutation

The crossover operator is mainly responsible for bringing diversity in the population; mutation operator is also used for bringing further diversity in the population to capture unique potential solution that might have missed in the initial population and prevents local optimum.Let the i[th] individual be noted as, $S_i(t)$ = [$S_1(t)$,……….$S_k(t)$………..$S_N(t)$], and the gene S_k be selected for mutation. The domain of the variable is given by $SK \in [x_i^{min}, x_i^{max}]$. The result of the mutation is

$$S_i(t) = [S_1(t) ,\ldots\ldots\overline{S_k}(t)\ldots\ldots S_N(t)]_i \quad (8)$$

where S_K is a random value within the domain of S_K according to the probability P_m in the order of 0.002 to 0.005.

Two mutation schemes are employed to determine$\overline{S_k}$; If$\overline{S_k}$ is a continuous variable like P_G or V, then $\overline{S_k}$ is a random value in the range$[S_k^{min}, S_k^{max}]$, $\overline{S_k}$ ϵ {PG, V}. And if $\overline{S_k}$ is a discrete variable like tap setting T_P and switchable shunt device Y_h, then $\overline{S_k}$ is a random value in the range [0, M_j], $\overline{S_k}$ ϵ {T_P, Y_h}.

3. Algorithm for OPF Implementation

The inequality constraints selected are generator bus upper voltage limits and lower voltage limits (0.95 to 1.10 pu fit generator bus and 0.950 to 1.05 pu for load bus) at every bus should be within the limits. Active power limits at generator buses ($P_{gi}^{min} \le Pgi \le P_{gi}^{max}$) and Reactive Power limits at generator buses and bus injections ($Q_{gi}^{min} \le Qgi \le Q_{gi}^{max}$) limits, Tap changing limits (0.9 to 1.1 pu), maximum loadability, slack bus power and size of shunts (0.0 to 0.05) within the specified limits. Violations in any of the active factors may adversely affect the system security and hence the limits on active factor s need to be adhered.

The fast decoupled load flow (FDLF) method is used to obtain the converged V and delta for each chromosome in the proposed algorithm andY_{bus} is formed by using sparsity. The technique is below.

Step 1 : acquire the system data and optimization parameters.
Step 2 : form ybus using sparsity technique, form constant slope matrix [B'] and decompose using cholesky decomposition.
Step 3 : model the discrete control variables (taps and shunts) and randomly generate the current population members with in their variable limits.
Step 4 : modify the elements of ybus due to taps and shunt elements and with the updated values of ybus elements form the slope matrix [B''] and decompose using cholesky decomposition.
Step 5 : run fdc power flow.
Step 6 : from converged load flow solution compute slack bus power, line losses, bus voltage magnitudes, phase angles.
Step 7 : initialize the penalty factors and calculate the penalty factors for violated functional constraints.
Step 8 : compute the objective function value and penalty factors are added if violation occurs.
Step 9 : check the fitness, apply elitism, roulette wheel technique and apply genetic operators.
Step 10: calculate individual generation of generators and corresponding fuel costs together with the total fuel cost, voltage profile, and total losses.

The proposed algorithm has been tested on a standard IEEE-30 bus system on a C2D computer of 2.1 GHz switching speed. The network consists of 6 Generator buses, 21 load buses and 41 lines, with a total load of 283.4 MW.24 control variables are selected in the Optimal Power flow problem. The gene length for unit active power outputs is 12 bits, generator voltage magnitude is 8 bits, and both of them are treated as continuous control variables. As the transformer tap settings can take 17 discrete values each one is encoded using 5 bits and the step size is 0.0125 p.u. The bus shunt susceptance can take 6 discrete values each one is encoded using 3bits, and the step size is 0.01 p.u. Thus, the total string length tends to be 155.

The established population size is 60 with a uniform crossover probability of 0.9. The string length is 155 bits, mutation and elitism probability of 0.05 and 0.2 respectively with a scaling factor of 0.6. PSO, the swarm size is 60 and the size of particle is 24. The acceleration constants are C1=C2=2.05, inertia Weight (W) is 1.2 and Constriction Factor (K) is 0.7295. For DE, the population size is 60, vector length is 24, Scaling factor of 0.8 with acrossover rate =0.7.

4. Results and Discussion

The Table 1 show the various control variables obtained using different optimization algorithms and Table 2 distinguishes and compares the algorithms in terms of the Fuel cost obtained, Losses and the time per iteration.

Table 1. Control variables with various optimization techniques

Control Variable	Base Case	ALGORITHM					
		SGA	PSO	DE	AGA	MBHS	PGA
Slack Bus	-	175.99	176.5	17.85	175.8	173.72	178.68
PG_2 (MW)	80.0	49.34	48.83	48.47	48.96	47.04	48.631
PG_5	50.0	21.93	21.13	20.78	22.01	23.4	21.936
PG_8	20.0	22.96	20.27	20.66	21.35	25.34	21.409
PG_{11}	20.0	12.78	12.37	10.00	10.96	10.67	10.00
PG_{13}	20.0	12.10	12.80	14.77	12.0	12.36	12.00
VG_1(pu)	1.0	1.05	1.05	1.05	1.05	1.05	1.05
VG_2	1.0	1.01	1.044	0.95	1.06	0.96	1.041
VG_5	1.0	1.09	1.043	1.09	0.99	1.055	1.035
VG_8	1.0	1.04	1.0	1.10	0.972	1.01	0.95
VG_{11}	1.0	1.08	1.02	1.05	1.02	0.98	0.95
VG_{13}	1.0	1.02	1.01	0.98	1.01	1.05	0.95
$T_{6,9}$(pu)	1.0	0.96	0.9	0.90	1.02	0.9	0.90
$T_{6,10}$	1.0	1.05	1.1	1.08	0.92	1.04	1.013
$T_{4,12}$	1.0	1.012	1.0	0.93	0.95	1.05	0.963
$T_{27,28}$	1.0	1.02	1.025	1.01	1.03	1.03	0.938
S_{10}(pu)	0.00	0.02	0.02	0.12	0.04	0.03	0.03
S_{12}	0.00	0.03	0.03	0.06	0.02	0.05	0.06
S_{15}	0.00	0.02	0.05	0.02	0.05	0.02	0.06
S_{17}	0.00	0.01	0.03	0.06	0.05	0.03	0.01
S_{20}	0.00	0.02	0.04	0.08	0.02	0.04	0.02
S_{21}	0.00	0.04	0.04	0.06	0.01	0.05	0.01
S_{23}	0.00	0.02	0.03	0.08	0.05	0.06	0.05
S_{24}	0.00	0.05	0.02	0.06	0.05	0.06	0.04
S_{29}	0.00	0.04	0.01	0.06	0.04	0.04	0.02

AGA : Adaptive GA
MBHS : Music Based Harmony Search
PGA : Proposed GA

Table 2. Comparison of algorithms in general in best optimized condition

TECHNIQUE	Fuel cost	Losses	Time /iter	Iterations
SGA (wo-VP)	802.359	9.60	0.488	40
PSO	802.667	9.59	1.737	70
DE	801.114	9.54	1.236	46
AGA	802.700	9.44	1.25	24
PGA	800.801	9.23	0.620	37

SGA-VP : Simple GA without Variable probability
PSO : particle swarm Optimization
DE : Differential Evolution

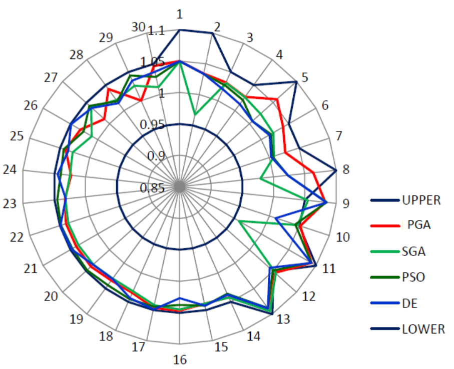

Figure 1. Voltage Profile of SGA, PSO, DE and PGA

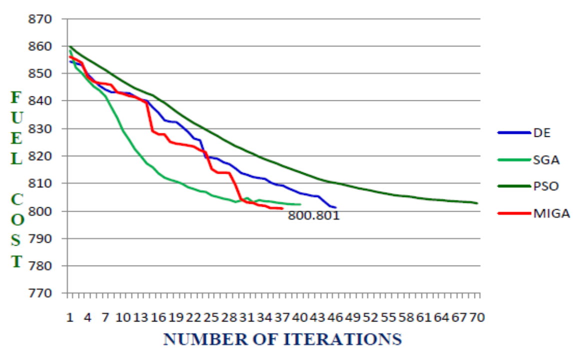

Figure 2. Convergence Characteristics of various algorithms

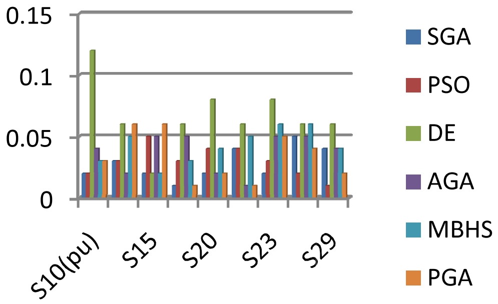

Figure 3. Shunt values using various algorithms

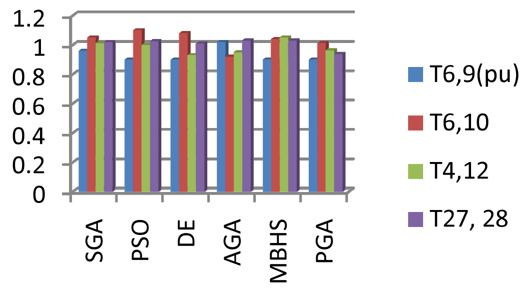

Figure 4. Tap settings using various algorithms

It's found that the proposed algorithm provides the lowest fuel cost of 800.801 dollars/per MW hr and is the second fastest algorithm next to simple genetic algorithm. Figure 1 illustrates the voltage profile obtained in various algorithms like SGA, PSO, DE and PGA. Figure 2 illustrates the Convergence characteristics, Figure 3 and Figure 4 demonstrates the Shunt and Tap settings in SGA, AGA, PSO, DE and PGA.

5. Conclusion

In this paper, a novel approach is developed which inherits the merits of binary coded and real-coded genetic algorithms. It's found that the hybrid method offers, the lowest fuel cost and is the second fast algorithm when compared to other variants of genetic algorithm like Simple Genetic algorithm, adaptive genetic algorithm, particle swarm optimization and differential evolution. The control parameters obtained by the proposed method confirms the robustness. The implementation has been performed on a standard IEEE-30 bus system and it's found that the proposed method is highly promising.

References

[1] Carpentier J. *Optimal Power Flows: Uses, methods and developments*. IFAC Symposium on planning and operation of electric energy systems. Rio de Janeiro. 1985.
[2] Pandya KS, Joshi SK. A survey of Optimal Power flow methods. *Journal of Theoretical and Applied Information Technology*. 2009; 4(5): 450-457.
[3] Gnanadass R, Venkatesh, Padhy N. Evolutionary Programming Based Optimal Power Flow for Units with Non-Smooth Fuel Cost Functions. *Electric Power Components and Systems*. 2004; 33(3): 349-361.
[4] Biskas P, Ziogos N, Tellidou A, Zoumas C, Bakirtzis A, Petridis V, Tsakoumis A. Comparison of two metaheuristics with mathematical programming methods for the solution of OPF. *IEE Proceedings on Generation, Transmission and Distribution*. 2006; 153(1): 16-24.
[5] Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. New York: Addison Wesley Publishing Company, Inc. 1989.
[6] Goldberg DE. *Sizing Populations for Serial and Parallel Genetic Algorithms*. Third International Conference on Genetic Algorithms. Morgan Kaufmann. 1989.
[7] Chaiyaratapiia N, Zalzala AMS. Recent Developments in Evolutionary and Genetic Algorithms: Theory and Applications. Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 97). Glasgow, UK. 1997: 270-276.
[8] Deb K. Multi-Objective Optimization using Evolutionary Algorithms. New York: John Wiley and Sons. 2001.
[9] Gaing ZL, Chang RF. *Security Constrained Optimal Power Flow by Mixed Integer Genetic Algorithm With Arithmetic Operators.* IEEE Power engineering society general meeting. Montreal, Que. 2006: 8.