Using Genetic Algorithm for Optimal Dispatching of Reactive Power in Power Systems

Robert Łukomski

Abstract Genetic algorithms are widely recognized as efficient tool for solution of complex and non-linear optimization problem of optimal reactive power dispatch and voltage control in power systems. The paper is addressed to the consideration of influence of different methods for generating initial population for genetic algorithm performance. Genetic algorithm operates on individuals representing some solutions. Randomly generated initial population evolve during the evolutionary process with use of some operations into the final population including probably the best task solution. The way of creating initial population decides on covering initial solution space. Hence, it may affect genetic algorithm performance. There exist variety of methods to generate members of initial population covering solution space. The paper presents an evaluation of pseudo-random numbers, gaussian and some space point process based algorithms to produce initial population in terms of the convergence speed and quality of the obtained optimization results.

Keywords Optimal reactive power dispatch · Genetic algorithm · Initial population · Random numbers

1 Introduction

The main objectives of reactive power (VAR) optimization are to improve node voltage profile and minimize active power losses under certain power system operating conditions. To achieve these goals power system operators perform control actions by adjusting generator voltages, on-load transformer tap changers

R. Łukomski (✉)
Faculty of Electrical Engineering, Wroclaw University of Technology, Wroclaw, Poland
e-mail: robert.lukomski@pwr.edu.pl

Z. Wilimowska et al. (eds.), *Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part IV*,
Advances in Intelligent Systems and Computing 432, DOI 10.1007/978-3-319-28567-2_16

(OLTC) and VAR generation/absorption by VAR sources located in power network (capacitor/reactor banks, static VAR sources etc.). It results in need for applying tools supporting operator decisions in real-time environments with continuously fluctuating and uncertain operation conditions.

VAR optimization is a complex non-linear and combinatorial optimization problem involving objective functions with multiple local extreme points and many non-linear constraints with not only continuous but also discrete decision variables. Originally, methods employing non-linear programming gradient based methods have been employed to solve this problem. However, these methods can stuck in local minima and have very limited capability of handling mixed discrete-continuous decision variables. Currently, the growing interest of artificial intelligence based optimization method has been observed. Genetic algorithms (GAs) and their variations (evolutionary programming, differential evolutions etc.) have been recognized as efficient tools for finding optima of objective functions.

Early attempts on using GAs to solve VAR optimization use modifications of simple binary GA which rely on problem decomposition and application of specialized genetic operators can be found [1]. To improve optimization results and efficiency GA are combined with other optimization methods such as interior point [2–4] and simulated annealing [5]. In [6] differential evolution method is recognized as fast and efficient tool for VAR dispatch. Real-coded GA exploiting special self-adaptive crossover and polynomial mutation scheme superior to evolutionary programming based optimization is given in [7]. Application of real coded GA for multi-objective optimization using Pareto-optimal set concept is reported in [8].

GA suffer from time consuming calculations and it will limit their applications in real-time operation. To improve numerical efficiency new evolution or hybrid schemes are developed. Theoretically, the convergence of the GA is independent of the initial points. Usually, for practical problems solution space is usually huge and algorithm can converge very slow to the solution. Some studies have revealed that the way the initial population is generated may influence the algorithm speed and quality of results [9, 10].

The problem of generating initial population for VAR optimization in power system has not been yet studied in details. Traditionally, initial population is created with use of the point pseudo-random generators available in programming environment. In this work other generation method of individuals, especially low discrepancy and gaussian are considered from viewpoint of result quality and convergence speed.

The paper is organized as follows. First, formulation of o VAR dispatching optimization problem is presented. Next, applied GA strategy is described in details. Numerical results of VAR dispatch optimization task concerning different ways on creating of initial population are shown. Some concluding remarks are presented at the end of this paper.

2 Formulation of VAR Dispatching Optimization Problem

The objective of VAR dispatching optimization is to minimize active power losses as a difference between generating and load power:

$$\min f(\mathbf{x},\mathbf{u}) = \Delta P = \sum_{k \in I_g} P_{G,k} - \sum_{l \in I_L} P_{L,l}, \tag{1}$$

where:

$\mathbf{x}$	vector of dependent variables,
$\mathbf{u}$	vector of decision variables,
ΔP	active power system losses,
$P_{G,k}$	active power generating at node k,
$P_{L,l}$	active power load at node l,
I_g, I_L	set of generating and load bus respectively.

Decision variable vector is constructed as follows:

$$\mathbf{u} = [\mathbf{V}_G \quad \mathbf{t}_p \quad \mathbf{Q}_{sh}]^T, \tag{2}$$

where:

$\mathbf{V}_G = [V_i]$, $i \in I_g$	row vector of voltages at generation buses and slack bus,
$\mathbf{t}_p = [t_j]$, $j \in I_{OLTC}$	row vector of discrete tap values of on-load tap changer transformers,
$\mathbf{Q}_{sh} = [Q_{sh,\ k}]$, $k \in I_{sh}$	row vector of values of shunt VAR,
I_{OLTC}	set of branches with OLTC,
I_{sh}	set of buses with adjustable reactive power shunts.

Depended variables values are obtained from power flow calculations and they are used for evaluation of solution feasibility i.e. preserving operational constraints:

$$\mathbf{x} = [\mathbf{V}_L \quad \mathbf{Q}_G \quad \mathbf{S}_{Br}]^T, \tag{3}$$

where:

$\mathbf{V}_G = [V_i]$, $i \in I_L$	row vector of voltages at load buses,
$\mathbf{Q}_G = [Q_{G,\ j}]$, $j \in I_g$,	row vector of reactive power at generation buses,
$\mathbf{S}_{Br} = [S_{Br,\ k}]$, $k \in I_{Br}$	row vector of apparent power branch flows,
I_{Br}	set of power system branches.

The equality constraints stem from power flow equations:

$$P_{G,i} - P_{L,i} - f_{P_i}(\mathbf{x}, \mathbf{u}) = 0 \tag{4}$$

$$Q_{G,i} - Q_{L,i} - f_{Q_i}(\mathbf{x}, \mathbf{u}) = 0 \tag{5}$$

where:

$f_{Pi}(\mathbf{x}, \mathbf{u})$, $f_{Qi}(\mathbf{x}, \mathbf{u})$	non-linear functions describing active and reactive powers at node i respectively,
P_{Gi}, Q_{Gi}	active and reactive power generations at node i respectively,
P_{Li}, Q_{Li}	active and reactive power loads at node i respectively.

Note that the GA is applied, the coded decision variables are self-constrained and their values are always in respective bounds.

Inequality constraints represent required power system operating constraints:

- generation VAR constraints:

$$Q_{G,i}^{\min} \le Q_{G,i} \le Q_{G,i}^{\max}, \quad i \in I_G, \tag{6}$$

- load bus voltage constraints:

$$V_i^{\min} \le V_i \le V_i^{\max}, \quad i \in I_L, \tag{7}$$

- branch flows constraints:

$$S_m \le S_m^{\max}, \quad m \in I_{Br}, \tag{8}$$

Note that the GA is applied, the coded decision variables are self-constrained and their values are always in respective bounds.

To handle the constraints (6)–(8) in fitness function f_f calculated for evaluation of individuals belonging to the GA population, the penalty quadratic terms corresponding to these inequalities are added:

$$f_f = \Delta P + \sum_{i \in I_L} \lambda_V \left(V_i - V_i^l\right)^2 + \sum_{i \in I_G} \lambda_{Qg} \left(Q_{gi} - Q_{gi}^l\right)^2 + \sum_{\substack{i,j \\ i \neq j}} \lambda_S \left(S_{ij} - S_{ij}^l\right)^2 \tag{9}$$

where: λ_V, λ_{Qg}, λ_S—penalty weighting coefficients and function in penalty terms are as follows:

$$X^l = \begin{cases} X, & X^{\min} \le X \le X^{\max} \\ X^{\min}, & X^{\min} > X \\ X^{\max}, & X^{\max} < X \end{cases} \qquad X \in \{V_i, Q_{gi}\}, \quad S_{ij}^l = \begin{cases} S_{ij}, & S_{ij} \le S_{ij}^{\max} \\ S_{ij}^{\max}, & S_{ij} > S_{ij}^{\max} \end{cases} \tag{10}$$

Fitness function value grows rapidly if some constrains are violated. In such way "bad" individuals can be marked and eliminated from population.

3 GA Based VAR Optimization Algorithm

The performance GA applied to VAR optimization problem is based on the crossover, mutation and selection. Some individuals belonging to the population are selected with use of assumed selection strategy, to form a set of parents. Parenting individuals are crossbred in pairs and offspring individuals are created. In addition, randomly selected individuals can be modified according to mutation schema. The best individuals form new population (elitism) and genetic operations are repeated in next iteration of GA called generation. The processing is continued until maximal number of generation is reached or solution cannot be further improved.

The applied GA use real-coding of decision variables instead of binary since it has been proven their better accuracy and speed in many optimization problem [11]. Instead of proportional selection very fast pairwise (binary) tournament selection here is employed. Many crossover and mutation operators for real-coded GA have been proposed. Here Single Binary Crossover (SBX) and polynomial mutation schemes described originally in [12] are applied.

The used GA requires setting some parameters: population size, tournament size, crossover and mutation probabilities, maximal number of generations, tolerance level within objective function value is not improved in last generations. These parameters were set by trial and error scheme with multiple running of algorithm to find the best choice.

The employed optimization algorithm contains the following steps:

1. Setting GA parameters.
2. Generating of initial population.
3. Calculation of fitness by (6) for each individual in current population.
4. Tournament selection for parenting individuals: random selection of two individuals and choosing the best one
5. Applying of parent individuals and applying of SBX. First, for the random number α the parameter is calculated as follows:

$$\beta = \begin{cases} (2\alpha)^{1/(\eta+1)}, & \alpha \in [0;0,5] \\ (2(1-\alpha))^{-1/(\eta+1)}, & \alpha \in (0,5;1] \end{cases}, \tag{11}$$

where: η—crossover parameter (real positive number).

The offspring individuals are calculated by formula:

$$x_i^{(t+1)} = \frac{1}{2}\left((1+\beta)x_i^{(t)} + (1-\beta)y_i^{(t)}\right), \quad y_i^{(t+1)} = \frac{1}{2}\left((1-\beta)x_i^{(t)} + (1+\beta)y_i^{(t)}\right), \tag{12}$$

6. Applying polynomial mutation with as:

$$y_i^{(t+1)} = x_i^{(t+1)} + \left(x_i^{\max} - x_i^{\min}\right)\delta_i, \tag{13}$$

where:

$\delta_i = \begin{cases} (2r_i)^{1/(\eta'-1)} - 1, & r_i < 0,5 \\ 1 - [2(1-r_i)]^{1/(\eta'+1)}, & r_i \geq 0,5 \end{cases}$, r_i—uniform random number, η'—mutation parameter (real positive number).

7. Checking stopping criteria. If the maximum generation number is reached or improving the solution cannot be obtained, stop the algorithm; otherwise go to step 3.

4 Generating Initial Population for GA

4.1 Coding of Decision Variables

As an initial population N_I individuals are created. In real-coded GAs, individuals are represented as N_D-dimensional real number vectors, where N_D is the dimension of the search space. For continuous decision variables the initial population is formed with use of the lower and upper bounds of decision variables as follows:

$$y_{ij} = y_j^{\min} + \left(y_j^{\max} - y_j^{\min}\right) x_{ij}, \tag{14}$$

where:

y_{ij}	value at *j*th position of *i*th individual,
$y_j^{\min}$, $y_j^{\max}$	lower and upper bonds for *j*th decision variable respectively,
$i = 1, 2, \ldots N_I, j = 1, 2, \ldots N_D, N_I$	population size,
N_D	number of continuous decision variables,
x_{ij}	random number generated from the range [0, 1].

Similarly, the discrete decision variables as transformer taps, are rounded-off to the nearest integer number:

$$y_{ij} = y_j^{\max} + \left(N_j^S\right)^{-1}\left(y_j^{\max} - y_j^{\min}\right)\left[N_j^S x_{ij}\right] \tag{15}$$

where:
$[x]$ rounding-off to the nearest integer of x,
N_j^s integer number of regulation steps of jth discrete decision variable.

Note that decision variables are self-constrained and their values are always in respective bounds. Hence initial population is generated by randomly distributed points in N-dimensional unity hyper-cube $\boldsymbol{I}^N$ belonging to $\boldsymbol{R}^N$ space.

4.2 Initial Population Generating Methods

A variety of methods do generate random sampling points have been proposed. Here, only four methods for initial population generation are evaluated:

- sampling based on subtract with borrow (SWB) pseudo-random uniform generator described in [13],
- Niderreiter quasi-random sampling (NQR) is dedicated to obtains numbers with maximal uniformity (low discrepancy) [14],
- non-aligned systematic (NAS) sampling use the transformation of initial pseudo-random number set to uniform distribution of sample points over decision variable hyper-space [9]. The unit hyper-cube formed this hyper-space by is divided into b^N elementary sub-cubes with one sample-point located in each sub-cube. For two dimensional space case the sample points representing the individuals are obtained by:

$$x = \left[\left((j-1) + r_{i,1}\right)D, \left((i-1) + r_{j,2}\right)D\right], \tag{16}$$

where:
$i, j = 1, 2, \ldots, b$, $D = 1/b$, b number of intervals,
$r_{i,1}$, $r_{j,2}$ elements of $b \times 2$ matrix $\mathbf{r}$ containing pseudo-random numbers.

An algorithm on generating points for n-dimensional hyper-space is given in [9]. Usually the number of sampling point considerably exceeds the number of required points, so in the next step they are selected randomly to match to required number,

- Gaussian sampling (GS) initializes population by samples from gaussian distribution with o mean $\mu^{(0)}$ and positive definite covariance matrix $\boldsymbol{\Sigma}^{(0)}$, so that

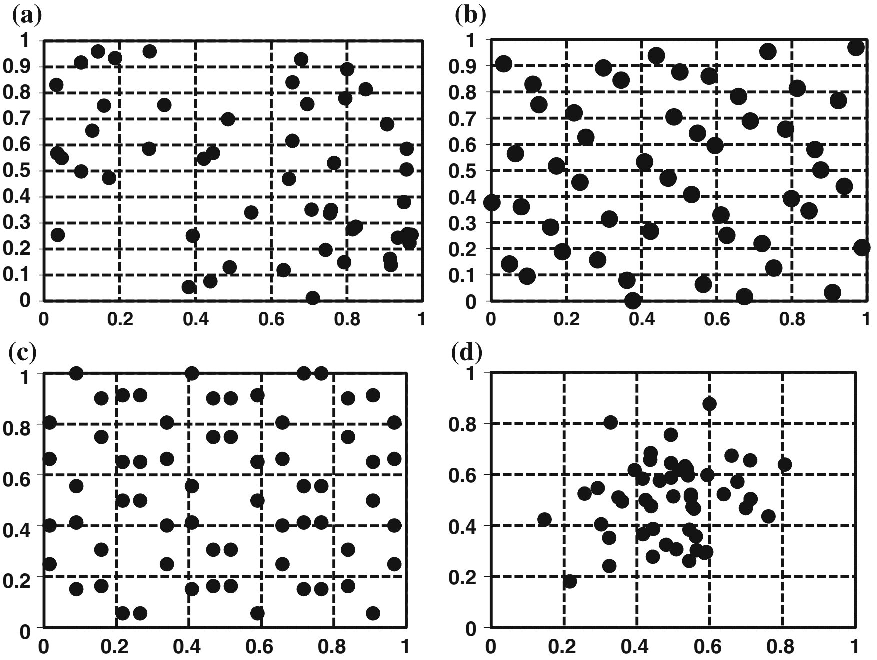

Fig. 1 Example multi-dimensional point sets mapped into 2D, obtained from various random generators: **a** SWB, **b** NQR, **c** NAS, **d** GS

each individual in the initial population is $\mathbf{X}^{(0)} \sim N(\mu^{(0)}, \Sigma^{(0)})$. Normally distributed random variables are assumed to be independent. It results in diagonal covariance matrix (non-diagonal elements are near zero). "Dispersion" of the sampling points depends on the value of variance. Note, that Gaussian sampling with large variance might be a quite good uniform sampling approximation.

Example distribution of sampling points generated by different methods for two-dimension space is shown in Fig. 1. Note that methods NQR and NAS give better space coverage than the remained ones.

5 Numerical Tests

5.1 General Assumptions

In order to investigate the influence of selected initial population generation methods on GA performance standard IEEE-30 bus system model has been used. Description of the system with detailed parameters can be found in [15].

Optimization problem was limited to 12 decision variables: 6 continuous generator voltages, 2 continues shunt values and 4 discrete tap changers with 20 regulation steps. Fitness function values were calculated with results obtained from accompanying fast electric network power flow program. All the numerical procedures and calculation steps were implemented in Mathworks Matlab environment.

5.2 GA Parameter Adjustment

GA parameters were determined by several trials with different parameter sets. Selected values for further investigations are shown in Table 1. Although maximal number of generation is set to 300, GA run can also be stopped before reaching max generation number, i.e. further best fitness function value cannot be improved. The tolerance level for best fitness is assumed to 10^{-5} (in last 10 generations). Parameters η and η' are both set to 20 accordingly to hints in [12].

The experiments were carried out as follows. For each initial population generation method 100 random sets with 600 (12 × 50) numbers were created. In order to handle the effect of the probabilistic nature of GA on results the algorithms are executed 30 times. Results comprising minimal and mean fitness function value, standard deviation, mean generation number until convergence condition is reached are shown in Table 2.

The mean of objective function is smallest for NAS sampling. However, the means of best objectives for different initial sampling strategies were compared with standard analysis of variance and there were no statistically significant differences detected. Overall improvement by reduction of the objective, i.e. active power losses with GA optimization by approximately is 4.00–4.65 %.

Similarly, the mean values (rounded to integer) of generation numbers $\overline{N_G}$ the convergence is obtained for different sampling methods are also comparable.

Table 1 Assumed GA parameter values

Parameter	Population size	Tournament size	Crossover rate	Mutation rate	Mutation size	Max generations
Value	50	2	0.8	0.8	0.1	300

Table 2 Summary of the obtained results

Parameter	Sampling method			
	SWB	NQR	NAS	GS
min ΔP	16.530	16.592	16.536	16.572
$\overline{\Delta P}$, MW	16.715	16.738	16.682	16.694
$\sigma_{\Delta P}$, MW	0.124	0.134	0.091	0.155
$\overline{N_G}$	150	129	157	151

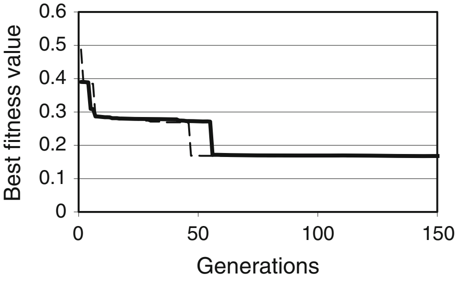

Fig. 2 Example GA convergence for SWB (*dotted line*) and NAS (*smooth line*)

However, NQR gives good solutions within less computation time. Example characteristics of convergence for different sampling methods are shown in Fig. 2.

Despite some differences, the results show that the random number generation strategies which are taken into consideration perform rather well if they are used in initial populations of genetic algorithms.

An important issue not discussed here is the size of the optimization problem. The more decision variables the more "sparse" is initial population, since the distance between adjacent points in initial population space increases even if good coverage is reached. It is expected that future research will better explain this problem.

6 Conclusions

In the paper different methods of generating initial population for GA applied to the practical optimization problem of voltage and VAR dispatch in example power system. This problem is recognized as complex optimization task with non-linear objective function and constraints and mixed nature of decision variables. Up to date works on creation of GA initial population topic suggest better performance of the GAs with population generated with low discrepancy random number sequences. However, they concentrate on testing with use of some theoretical benchmark functions. This attempt concentrate on selected practical optimization problem.

The obtained results reveal that the standard way to generate initial population with SWB pseudo-random generator with uniform sampling is comparable to the others low-discrepancy and gaussian sampling methods. Convergence speed and mean objective functions values are comparable in statistical sense. It should be pointed out that using NQR, NAS sampling usually results in a little faster convergence in first generations of GA run.

Future works will concentrate on testing of other methods for generating GA initial population, especially methods forming clusters resulting from prior knowledge on solution space. Different genetic algorithm parameters like population size and maximum number of generations will also be considered. Experience with VAR dispatch optimization will also be applied to optimize VAR dispatching in actual size power systems with size of several hundred buses.

References

1. Iba, K.: Reactive power optimization by genetic algorithm. IEEE Trans. Power Syst. **9**(2), 685–692 (1994)
2. Wu, Q.H., Ma, J.T.: Power system optimal reactive power dispatch using evolutionary programming. IEEE Trans. Power Syst. **10**(3), 1243–1249 (1995)
3. Yan, W., Liu, F., Chung, C.Y., Wong, K.P.: A hybrid genetic algorithm-interior point method for optimal reactive power flow. IEEE Trans. Power Syst. **21**(3), 1163–1169 (2006)
4. Yan, W., Lu, S., Yu, D.C.: A novel optimal reactive power dispatch method based on an improved hybrid evolutionary programming technique. IEEE Trans. Power Syst. **19**(2), 913–918 (2004)
5. Das, D.B., Patvardhan, C.: A new hybrid evolutionary strategy for reactive power dispatch. Electr. Power Syst. Res. **65**(2), 83–90 (2003)
6. Liang, C.H., Chung, C.Y., Wong, K.P., Duan, X.Z., Tse, C.T.: Study of differential evolution for optimal reactive power flow. IET Gen. Transm. Disrib. **1**(2), 253–260 (2007)
7. Subbaraj, P., Rajnarayanan, P.N.: Optimal reactive power dispatch using self-adaptive real coded genetic algorithm. Electr. Power Syst. Res. **79**(2), 374–381 (2009)
8. Zhihuan, L., Yinhong, L., Xianzhong, D.: Improved strength Pareto evolutionary algorithm with local search strategies for optimal reactive power flow. Inf. Technol. J. **9**(4), 749–757 (2010)
9. Maaranen, H., Miettinen, K., Penttinen, A.: On initial populations of a genetic algorithm for continuous optimization process. J. Global Optim. **37**, 405–436 (2007)
10. Kimura, S., Matsumura, K.: Genetic algorithms using low-discrepancy sequences. Proceedings of the Genetic and Evolutionary Computation Conference, New York, pp. 1341–1346 (2005)
11. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolutionary Programs, Springer, Berlin (1996)
12. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley Press, Hoboken, pp. 113–125 (2002)
13. Marsaglia, G., Zaman, A.: A new class of random number generators. Ann. Appl. Prob. **1**(3), 462–480 (1994)
14. Niederreiter, H.: Low-discrepancy and low-dispersion sequences. J. Number Theory **30**, 51–70 (1988)
15. Lee, K., Park, Y., Ortiz, J.: A united approach to optimal real and reactive power dispatch. IEEE Trans. PAS-104, pp. 1147–1153 (1985)