Application of ANN, SI and EC in power system

• **ANN** – adaptive neurocontroller
• **PSO** – optimal PI controller parameters for the DFIG rotor-side converter
• **GA** – Solving the Optimal Power Flow Problem
ANN – Adaptive Neurocontroller

“A Continually Online Trained Neurocontroller for Excitation and Turbine Control of a Turbogenerator”
Ganesh K. Venayagamoorthy and Ronald G. Harley
IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 16, NO. 3, SEPTEMBER 2001

**Background:** The increasing complexity of the modern power grid highlights the need for advanced modeling and control techniques for effective control of turbogenerators.

Continually Online Trained (COT) Artificial Neural Network (ANN):
*neuroidentifier*, to identify the complex nonlinear dynamics of the power system
*neurocontroller*, to control the turbogenerator → replaces the conventional automatic voltage regulator (AVR) and turbine governor
Turbogenerator & Conventional Automatic Voltage Regulator (AVR)

Turbogenerators:
• Supply most of the electrical energy and their performance is directly related to the stability and security of the power grid
• Characteristics: nonlinear, nonstationary and multivariable system

Conventional automatic voltage regulators and turbine governors:
• They were designed based on some linearized power system model to control the turbogenerator in some optimal fashion around one operating point
• At other operating points the control performance degrades → For large disturbances, these controllers operate outside the linear range and performance also degrades, thus driving the power system into undesirable operating states.
The necessity of supervisory control

Conventional Controllers for Turbogenerators:
* Mostly adaptive controllers → linear models, with certain assumptions of types of noise and possible disturbances.

However, the turbogenerator system is nonlinear, with complex dynamic and transient processes, hence it cannot be completely described by such linear models.

Moreover, for the design of adaptive controllers:
Assumptions: the number of system inputs equals the number of system outputs → to reduce the dimensions of the output space, with the drawback that this degrades the description of the system dynamics.

supervisory control: ANN
* good at identifying and controlling nonlinear systems.
* suitable for multivariable applications
* able to identify the complex and nonlinear dynamics problems with sufficient accuracy,
A micro-alternator, driven by a dc motor whose torque–speed characteristics are controlled by a power electronic converter to act as a micro-turbine, and a single short transmission line which links the micro-alternator to an infinite bus.
A 3 kW, 220 V, three phase micro-alternator

Conventional AVR and exciter combination

Transfer function:

**TABLE I**
**MICRO-ALTERNATOR PARAMETERS**

<table>
<thead>
<tr>
<th>$T_{q0}'$</th>
<th>$T_{q0}''$</th>
<th>$X_{q0}'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.69 s</td>
<td>0.25 s</td>
<td>0.205 pu</td>
</tr>
<tr>
<td>$T_{d0}'$</td>
<td>$T_{d0}''$</td>
<td>$X_{d0}'$</td>
</tr>
<tr>
<td>0.66 s</td>
<td>27 ms</td>
<td>0.164 pu</td>
</tr>
<tr>
<td>$T_{d0}''$</td>
<td>$T_{kd}$</td>
<td>$X_{q}$</td>
</tr>
<tr>
<td>33 ms</td>
<td>38 ms</td>
<td>1.98 pu</td>
</tr>
<tr>
<td>$T_{d}$</td>
<td>$X_{d}$</td>
<td>$X_{d}''$</td>
</tr>
<tr>
<td>26.4 ms</td>
<td>2.09 pu</td>
<td>0.213 pu</td>
</tr>
</tbody>
</table>

**TABLE II**
**AVR AND EXCITER TIME CONSTANTS**

<table>
<thead>
<tr>
<th>$T_{v1}$</th>
<th>$T_{v2}$</th>
<th>$T_{v3}$</th>
<th>$T_{v4}$</th>
<th>$T_{v5}$</th>
<th>$T_{e}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.616 s</td>
<td>2.266 s</td>
<td>0.189 s</td>
<td>0.039 s</td>
<td>0.0235 s</td>
<td>0.47 s</td>
</tr>
</tbody>
</table>

$T_{v1}$, $T_{v2}$, $T_{v3}$, $T_{v4}$: time constants of the PID voltage regulator compensator
$T_{v5}$: input filter time constant
$T_{e}$: exciter time constant
$K_{av}$: AVR gain
$V_{fdm}$: exciter ceiling
$V_{ma}$: AVR maximum
$V_{mi}$: AVR minimum

$V_{v1}$: Input Filter
$V_{v2}$: PID Compensation and limits
$V_{v3}$: AVR
$V_{v4}$: Exciter
$V_{v5}$: Saturation
$V_{v6}$: Transfer function
$V_{v7}$: Output
Micro-turbine: A separately excited 5.6 kW dc motor to drive the micro-alternator

Transfer function:

\[
\Delta \omega = \frac{K_p (1+sT_{g1})}{s + \frac{g_2}{1+sT_{g2}}} \Delta P + \text{saturation} \rightarrow \text{servo motor} \rightarrow \text{entained steam} \rightarrow \text{reheater} \rightarrow \frac{1}{1+sT_{g3}} \rightarrow \frac{1}{1+sT_{g4}} \rightarrow \frac{1}{1+sT_{g5}} \rightarrow P_m
\]
DESIGN OF THE NEUROCONTROLLER

Neuroidentifier Architecture

Using series-parallel Nonlinear Auto Regressive Moving Average (NARMA) model. This model output at time depends on both past values of output and past values of input.

\[ \hat{y}(t + 1) = f \left[ y(t), y(t - 1), \ldots, y(t - n + 1), \right. \]
\[ \left. u(t), u(t - 1), \ldots, u(t - m + 1) \right] \]

where \( y(t) \) and \( u(t) \) represent the output and input of the plant to be controlled at time \( t \), respectively.

The reason of using NARMA model:

online training is desired to correctly identify the dynamics of the turbogenerator → therefore avoiding a feedback loop in the model, which allows static backpropagation to be used to adjust the neural network weights (This reduces the computational overhead substantially for online training)
Adaptive Neurocontroller
DESIGN OF THE NEUROCONTROLLER...cont

Neuroidentifier Structure

12 inputs: time delayed by a sample period of 20ms

- Actual speed deviation of generator
- Actual terminal voltage deviation
- Actual deviation in the input to the turbine
- Actual deviation in the input to the exciter

Hidden layer with 14 neurons

- $\Delta \omega(t-1)$
- $\Delta \omega(t-2)$
- $\Delta \omega(t-3)$
- $\Delta V_f(t-1)$
- $\Delta V_f(t-2)$
- $\Delta V_f(t-3)$
- $\Delta P_{ref}(t-1)$
- $\Delta P_{ref}(t-2)$
- $\Delta P_{ref}(t-3)$
- $\Delta V_e(t-1)$
- $\Delta V_e(t-2)$
- $\Delta V_e(t-3)$

Neural network diagram:

- Estimated speed deviation $\hat{\Delta \omega(t)}$
- Estimated terminal voltage deviation $\hat{\Delta V_f(t)}$

One step ahead estimated terminal voltage deviation
DESIGN OF THE NEUROCONTROLLER...cont

**Neurocontroller Architecture**

Actual speed and terminal voltage deviation (each of these inputs is time delayed by 20ms)

- \( \Delta \omega(t-1) \)
- \( \Delta \omega(t-2) \)
- \( \Delta \omega(t-3) \)
- \( \Delta V_e(t-1) \)
- \( \Delta V_e(t-2) \)
- \( \Delta V_e(t-3) \)

**Hidden layer with 10 neurons**

- Deviation in the field voltage \( \Delta V_e \)
- Deviation in the power signal \( \Delta P_{ref} \)

The and the deviation in the power signal augment
Design of the Neurocontroller...cont

**Desired Response Predictor Characteristics:**

- It must be flexible enough to modify the dynamic performance of the neurocontroller such as the rise time and damping.
- The desired response signal must ensure that the turbogenerator is inherently stable at all times. In other words, the predictor must be stable.
- The desired response signal must incorporate the effects of a power system stabilizer.
Training process

The training of the **neuroidentifier** and **neurocontroller** takes place in two phases: *pre-control phase* and *post-control phase*.

**Pre-control phase**
S1 and S2 are in position 1

**Neuroidentifier Pre-training**

The neuroidentifier starts off with random initial values for its weights, but after about 3 seconds the weights have adjusted so that the estimated speed deviation is from the neuroidentifier correctly tracks or identifies the actual speed deviation.

The neurocontroller and neuroidentifier accept measurements from the turbogenerator and use this for training; *not yet control the turbogenerator*.
Training process...cont.

Pre-training of the neurocontroller, takes place with the trained neuroidentifier in cascade + Desired Response Predictor.

Output of the neuroidentifier at E, and the desired output at M, are subtracted to produce an error signal at H, which is backpropagated through the neuroidentifier (without changing its weights) to produce $\Delta u(k)$ at J. The error $K$ between $\Delta u(k)$ and $\Delta u(k)$ from the neurocontroller, uses backpropagation to change the neurocontroller weights in a direction which drives $K$ to zero, and therefore $H$ to zero.

Neuroidentifier weights:
- Fixed
- Backpropagation of errors at H

TDL: tapped delay line
First Procedure: Training the Neuroidentifier

a) The terminal voltage and speed deviations from their set points, are sampled at D and are time delayed by one, two and three sample periods.

b) The signals from a) are input at A to the neurocontroller which then calculates the signals $\Delta u(k)$ to be used to train the neuroidentifier.

c) These damping signals $\Delta u(k)$ are time delayed by one, two and three sample periods, and, together with the signals from a), are input to the neuroidentifier at G and C, respectively.

d) The output of the turbogenerator at D, and the output of the neuroidentifier at E, are subtracted to produce a first error signal F which, via backpropagation, is used to update the weights in the neuroidentifier.
Training process...cont.

e) The output of the neuroidentifier at E, and the desired output at M are subtracted to produce a second error signal at H.
f) The error signal at H from step e) is backpropagated at I, through the neuroidentifier, and obtained at J without changing the weights in the neuroidentifier.
g) The backpropagated signals J from f) are subtracted from the output signals of the neurocontroller to produce a third error signal at K.
h) This signal at K is then used to update the weights in the neurocontroller, using the backpropagation algorithm. This causes the neurocontroller to change its output in a way which drives the error signal K to zero.

Second Procedure: Training the Neurocontroller

Third Procedure: Controlling the turbogenerator
New control signals $\Delta u$ are calculated using the updated weights from the second procedure and are applied at time (k+1) to the turbogenerator at B.
Simulation results

±5% Step Changes in the Desired Terminal Voltage Setpoint

\[ P = 1.0 \text{ pu} & \quad Q = 0.62 \text{ pu}, \]
\[ Z = 0.02 + j 0.4 \]

\[ P = 1.0 \text{ pu} & \quad Q = 0.62 \text{ pu}, \]
\[ Z = 0.025 + j 0.6 \]
Simulation results

Three Phase Short Circuit at the Infinite Bus

- The short circuit test is carried out at: \( Z = 0.025 + j 0.6 \) at \( P = 1 \text{ pu} \) & \( Q = 0.62 \text{ pu} \)
Summary

The superior performance of the neurocontroller occurs because the online training never stops and deviation signals are used.

Another important consideration is that the neural networks have no prior information of the turbogenerator and the grid it is connected to, need no tuning on site during commissioning, and are therefore completely self-commissioning.

Such neurocontrollers allow power plants to be operated closer to their steady state stability limits, thus producing more electrical power per Dollar invested.

Further studies are in progress with this neurocontroller on a threemachine power system which suffers from multi-mode oscillations, and preliminary results look encouraging.
PSO – optimal PI controller parameters for the DFIG rotor-side converter

“Design of Optimal PI Controllers for Doubly Fed Induction Generators Driven by Wind Turbines Using Particle Swarm Optimization”
Wei Qiao, Ganesh K. Venayagamoorthy, Ronald G. Harley
2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
Introduction

Wind generation: the leading source among various renewable energy sources in the power industry

Wind turbine generator configurations:
* Fixed-speed wind turbine equipped with a conventional induction generator
* Variable-speed wind turbine equipped with a synchronous generator
* Variable-speed wind turbine equipped with a doubly fed induction generator (DFIG)

Variable-speed wind turbine (VSWT) equipped with a DFIG → attract attention due to its noticeable advantages:
* The concept: the wound rotor induction generator is grid-connected at the stator terminals, but the rotor terminals are connected to the grid via a partial-load variable frequency AC/DC/AC converter (VFC) and a transformer
* The VFC only needs to handle a fraction (25-30%) of the total power to achieve full control of the generator
### DFIG Type of Wind Turbine

**Advantages of VSWT with a DFIG – compared to wind turbine equipped with fixed-speed induction generator:**

- Provide decoupled control of active and reactive power of the generator
- More efficient energy production
- Improved power quality
- Improved dynamic performance during power system disturbances such as network voltage sags and short circuit (improved grid fault ride-through capability)

**Advantages of VSWT with a DFIG – compared to VSWT equipped with a synchronous generator** — a full load VFC is connected directly between the generator stator and the grid—

***The VFC of the DFIG is smaller in size and therefore much cheaper***

**About the VFC and its power electronics (IGBT switches):**

- Small power capacity
- Very sensitive to grid faults, easily destroyed by over-current
- The rotor-side converter might be blocked due to the protection from the over-current in the rotor circuit
- The wind turbine might be tripped from the system

**Significance of the control system:** *well-designed control system improves the grid fault ride-through capability of the wind turbine*

***Linear PI controllers are used to design the DFIG control system***
This study used particle swarm optimization (PSO) method to determine the optimal parameters of the four PI controllers for the DFIG rotor-side converter.

Results: optimal PI controllers enhances the transient performance and fault ride-through capability of the wind turbine generator.
The wound-rotor induction machine in this configuration is fed from both stator and rotor sides. In order to produce electrical power at constant voltage and frequency to the utility grid over a wide operation range from subsynchronous to supersynchronous speed, the power flow between the rotor circuit and the grid must be controlled both in magnitude and in direction.

The crow-bar is used to short-circuit the RSC in order to protect the RSC from over-current in the rotor circuit during grid faults.

In order to produce electrical power at constant voltage and frequency to the utility grid over a wide operation range from subsynchronous to supersynchronous speed, the power flow between the rotor circuit and the grid must be controlled both in magnitude and in direction.

VFC consists of two four-quadrant IGBT PWM converters connected back-to-back by a dc-link capacitor.
Aggregated model: Hundreds of individual wind turbines and DFIGs in a wind farm are modeled as an equivalent DFIG driven by a single equivalent wind turbine.
Control objective: regulate both the stator-side active power (by means of speed control) and reactive power (by means of rotor current regulation) independently.

* Four proportional gains: $K_\omega$, $K_Q$, $K_d$, and $K_q$
* Four integral time constants: $T_\omega$, $T_Q$, $T_d$, and $T_q$

Vector Control Scheme of the RSC
Control objective: to keep the dc-link voltage constant regardless of the magnitude and direction of the rotor current
Design of Optimal PI Controllers

**Objective:** determine optimal parameters of the RSC PI controllers, which reduces the over-current in the rotor circuit and the RSC during grid faults and therefore improve the transient performance of the DFIG

**Solution:** Using the PSO to find out the optimal parameters of the four PI controllers, which are four proportional gains ($K_\omega, K_Q, K_d,$ and $K_q$) and four integral time constants ($T_\omega, T_Q, T_d,$ and $T_q$), to optimize some performance measure function (fitness function)

The PI controller performance in the time domain can be measured by a set of parameters:

(i) Overshoot $M_p$
(ii) Rise time $t_r$
(iii) Settling time $t_s$
(iv) Steady-state error $E_{ss}$
Design of Optimal PI Controllers (2)

Performance measure function:

\[ f(x) = \beta \cdot \Delta I_{r,max} + (1 - \beta)(t_s - t_0) + \alpha \cdot |E_{ss}| \]

\[ x = [K_\omega, K_Q, K_{dr}, K_{dq}, T_\omega, T_Q, T_{dr}, T_{dq}] \]

\( \beta \) and \( \alpha \) : weighting factors to satisfy different design requirements

\( \Delta I_{r,max} \): maximum rotor current magnitude deviation of the DFIG

\( t_0 \): starting time of the disturbance

\( t_s \): settling time

If a large value of \( \beta \) is used, then the objective is to reduce the over-current in the rotor circuit. If a small value of \( \beta \) is used, then the objective is to reduce the settling time. The weighting factor \( \alpha \) is introduced to minimize the steady-state error.

Initial design: conventional methods - bode design and pole-zero placement at a specific operating condition

The parameters of the RSC PI controllers are then optimized by the PSO algorithm to minimize the value of the performance measure function.
Design of Optimal PI Controllers (3)

Flowchart of the optimal PI controller parameters design procedure

Five particles (N=5) are used in the simulation and the position vector of the first particle is initialized as the initially designed parameters; while the position vectors of the other four particles are initialized with the values around the initially designed parameters.

The values of $c_1$ and $c_2$ are chosen as 2; the inertia constant ($w$) = 0.8. The weighting factors are chosen as $\alpha = 0$, $\beta = 1$ in order to limit the over-current in the rotor circuit during grid faults.
The PSO is implemented with 30 trial runs by applying a 100 ms three-phase short circuit at the receiving end of line 2.

### INITIAL AND OPTIMAL PARAMETERS OF THE RSC CONTROLLERS

<table>
<thead>
<tr>
<th></th>
<th>$K_\omega$</th>
<th>$K_Q$</th>
<th>$K_d$</th>
<th>$K_a$</th>
<th>$T_\omega$</th>
<th>$T_Q$</th>
<th>$T_d$</th>
<th>$T_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial design</td>
<td>8.48</td>
<td>0.01</td>
<td>2.89</td>
<td>1.79</td>
<td>0.081</td>
<td>2.0</td>
<td>0.028</td>
<td>0.065</td>
</tr>
<tr>
<td>Optimal design</td>
<td>18.23</td>
<td>0.001</td>
<td>3.05</td>
<td>4.87</td>
<td>0.038</td>
<td>1.0</td>
<td>0.056</td>
<td>0.21</td>
</tr>
</tbody>
</table>
Case I: a 100 ms three-phase short circuit test at receiving end of line 2

- Initial design: rotor current magnitude is 19 kA
- Optimal design: rotor current magnitude is limited to 14 kA
Case II: a three-phase short circuit test at sending end of line 2, line 2 is tripped off from the system 100 ms after applying the fault
✓ Initial design: rotor current magnitude is 24 kA
✓ Optimal design: rotor current magnitude is limited to 13.5 kA
A Genetic Algorithm for Solving the Optimal Power Flow Problem
Tarek BOUKTIR, Linda SLIMANI, M. BELKACEMI
http://ljs.academicdirect.org/A04/44_58.htm

The objective is to minimize the fuel cost and keep the power outputs of generators, bus voltages, shunt capacitors/reactors and transformers tap-setting in their secure limits.

The OPF has been usually considered as the minimization of an objective function representing the generation cost and/or the transmission loss.

Effective optimal power flow is limited by:
(i) the high dimensionality of power systems
(ii) the incomplete domain dependent knowledge of power system engineers

Conventional methods:
* limited by numerical optimization procedures based on successive linearization using the first and the second derivatives of objective functions and their constraints as the search directions or by linear programming solutions to imprecise models
* the sensitivity to problem formulation, algorithm selection and usually converge to a local minimum
* incomplete domain knowledge, precludes also the reliable use of expert systems where rule completeness is not possible
Introduction...(cont.)

GA offers:
* the increasing availability of high performance computers at relatively low costs
* Extensive applications in solving global optimization searching problems when the closed-form optimization technique cannot be applied.
* Parallel and global search techniques that emulate natural genetic operators
* more likely to converge toward the global solution because it simultaneously, evaluates many points in the parameter space.
* need to assume that the search space is differentiable or continuous

Genetic Algorithms:
● The algorithms work with a population of string, searching many peaks in parallel, as opposed to a single point.
● work directly with strings of characters representing the parameters set not the parameters themselves.
● use probabilistic transition rules instead of deterministic rules.
● use objective function information instead of derivatives or others auxiliary knowledge.
● have the potential to find solutions in many different areas of the search space simultaneously.
The **objective function** for the entire power system can then be written as the sum of the quadratic cost model at each generator.

\[
F(x) = \sum_{i=1}^{ng} \left( a_i + b_i P_{gi} + c_i P_{gi}^2 \right)
\]

- \( ng \): number of generation including the slack bus
- \( P_{gi} \): generated active power at bus \( i \)
- \( a_i, b_i, c_i \): unit costs curve for \( i^{th} \) generator

**Types of equality constraints**: while minimizing the cost function, it is necessary to make sure that the generation still supplies the load demands plus losses in transmission lines.

**The power flow equations are used as equality constraints**

\[
\begin{bmatrix}
\Delta P_i \\
\Delta Q_i
\end{bmatrix} = \begin{bmatrix}
P_i(V, \theta) - (P_{gi} - P_{di}) \\
Q_i(V, \theta) - (Q_{gi} - Q_{di})
\end{bmatrix} = 0
\]

- \( P_i(V, \theta) = \sum_{j=2, nbus} V_{ij} \left( g_{ij} \cos \theta_j + b_{ij} \sin \theta_j \right); i = 2, nbus \)
- \( Q_i(V, \theta) = \sum_{j=2, nbus} V_{ij} \left( g_{ij} \sin \theta_j + b_{ij} \cos \theta_j \right); i = npv + 1, nbus \)

**Active and reactive power injection at bus \( i \) are defined in the following equation**
Problem Formulation (2)

Types of inequality constraints: reflect the limits on physical devices in the power system as well as the limits created to ensure system security

* Upper and lower bounds on the active generations at generator buses
  \[ P_{g_i}^{\text{min}} \leq P_{g_i} \leq P_{g_i}^{\text{max}}, \quad i = 1, \ldots, \text{ng}. \]

• Upper and lower bounds on the reactive power generations at generator buses and reactive power injection at buses with VAR compensation:
  \[ Q_{g_i}^{\text{min}} \leq Q_{g_i} \leq Q_{g_i}^{\text{max}}, \quad i = 1, \ldots, \text{npv}. \]

• Upper and lower bounds on the voltage magnitude at the all buses:
  \[ V_i^{\text{min}} \leq V_i \leq V_i^{\text{max}}, \quad i = 1, \ldots, \text{nbus}. \]

• Upper and lower bounds on the bus voltage phase angles:
  \[ \delta_i^{\text{min}} \leq \delta_i \leq \delta_i^{\text{max}}, \quad i = 1, \ldots, \text{nbus}. \]
GA applied to OPF

The active generation power for 9 bus IEEE system ($P_{g1}, P_{g2}$ and $P_{g3}$) would be coded as binary string of 0’s and 1’ with length $B_1$, $B_2$ and $B_3$, respectively. Each parameter $P_{gi}$ have upper bound $U_i$ (max) and lower bound $L_i$ (min). The choice of $B_1$, $B_2$ and $B_3$ for the parameters is concerned with the resolution specified by the designer in the search space. In the binary coding method, the bit length $B_i$ and the corresponding resolution $R_i$ is related by:

$$R_i = \frac{U_i - L_i}{2^{B_i} - 1}$$

If the resolution $(R_1, R_2, R_3)$ is specified as $(0.1, 0.05, 0.1)$, from (3) we have $(B_1, B_2, B_3) = (4, 4, 4)$.

If the candidate parameters set is $(1.7, 0.30, 1.1)$, then the chromosome is a binary string $111000110111$. The decoding procedure is the reverse procedure.
• The first step of any genetic algorithm is to generate the initial population.
  • A binary string of length L is associated to each member (individual) of the population.
  • The string is usually known as a chromosome and represents a solution of the problem.
  • A sampling of this initial population creates an intermediate population.
  • Thus, some operators (reproduction, crossover and mutation) are applied to this new intermediate population in order to obtain a new one.

Process starts from the present population and leads to the new population → a generation when executing a genetic algorithm.
A pair of parents selected from the population the recombination operation divides two strings of bits into segments by setting a crossover point at random, \textit{Single Point Crossover}.

The strings to be crossed are selected according to their scores using the \textit{roulette wheel}. Thus, the strings with larger scores have more chances to be mixed with other strings because all the copies in the roulette have the same probability to be selected.
GA applied to OPF (4)

- **Mutation** is a secondary operator and prevents the premature stopping of the algorithm in a local solution.
- The **mutation operator** is defined by a random bit value change in a chosen string with a low probability of such change.
- The mutation is necessary to avoid very similar all possible solutions, after some generations.
- All strings and bits have the same probability of mutation.

For example, in the string 110011101101, if the mutation affects to time bit number six, the string obtained is 110011001101 and the value of Pg2 change from 0.85 p.u to 0.75 p.u.
GA applied to OPF (5)

**Reproduction** is based on the principle of survival of the better fitness. It is an operator that obtains a fixed number of copies of solutions according to their fitness value. If the score increases, then the number of copies increases too.

The cost function is defined as:

\[ F(x) = \sum_{i=1}^{N} \left( a_i + b_i P_{g_i} + c_i P_{g_i}^2 \right), P_{g_i}^{\text{min}} \leq P_{g_i} \leq P_{g_i}^{\text{max}} \]

Our objective is to search \((P_{g_1}, P_{g_2}, P_{g_3})\) in their admissible limits to achieve the optimization problem of OPF.

The value of the cost is then mapped into a fitness value \(\text{Fit}(P_{g_1}, P_{g_2}, P_{g_3})\).

To minimize \(F(x)\) is *equivalent* to getting a maximum fitness value in the searching process. A chromosome 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 in the simulated roulette wheel as follows:

\[
\text{fitness}_i = \begin{cases} 
  f_{\text{max}} - f_i & \text{if } f_{\text{max}} \leq f_i; i = 1, n_g \\
  0 & \text{otherwise}
\end{cases}
\]
The summarized diagram of GA procedure

\[ F_i = \sum_{i=1}^{n} (a_i + b_i P_{g_i} + c_i P_{g_i}^2) \]

\[ \text{fitness} = \begin{cases} f_{\text{max}} - f_i & \text{if } f_{\text{max}} \leq f_i \\ 0 & \text{otherwise} \end{cases} \]