Chapter 4

Implementation of Genetic Algorithm based Real Power Dispatch

Introduction

The operation planning of a power system is characterized by having to maintain a high degree of economy and reliability. Economic load dispatch is one of the options available to the engineer to operate the system. Various mathematical programming methods and optimization techniques have been applied to solve this problem. These methods include lambda-iteration, the base point and participation factor, gradient based, unit-based genetic algorithm, evolutionary algorithm, etc. In this chapter, an innovative approach based on the lambda-based encoding GA to solve the problem of real power dispatch incorporating the branch power flow limits as described in chapter two is presented. In this approach, the normalized system incremental cost (lambda) was encoded within limits as determined by the minimal and maximal incremental cost of all actually synchronized units. By doing so, the number of bits required for coding was drastically reduced, since the number of bits of chromosome required is independent of the number of units. This feature therefore makes it attractive for large scale problems. Because the lambda based encoding GA approach was derived from the principle of well known classical technique, it was most expedient to first implement the classical approach for comparison with the new GA approach. Also unit based encoding GA real power dispatch was equally implemented. Each of the three methods are first sketched highlighting their principles of operation. The three methods were implemented and relatively investigated on the operator training simulator used in this work, and the preliminary results compared and conclusions drawn.

4.1 Classical Economic Load Dispatch

The economic load dispatch problem described in chapter 2 is a constrained optimization and has been widely solved using the LaGrange function. The basic principle associated with this method is that for continuous analytical generator cost functions, the most economical division of load between two or more units is when they are operating at equal incremental cost λ in CU/MWh subject to their capabilities constraints. This is formed by augmenting the objective function with the penalty term of violating the constraint function. The LaGrange function is given by:

$$\ell = F_T + \lambda\varphi \tag{4.1}$$

where F_T is as given by equation (2.10) and φ is as given by equation (2.11) in chapter 2.

The necessary conditions for an extreme value of the objective function result by taking the partial derivative of equation (4.1) with respect to generating units power output and set the derivatives equal to zero, we have

$$\frac{\partial \ell}{\partial P_{Gi}} = \frac{dF_i}{dP_{Gi}} - \lambda\left(1.0 - \frac{\partial P_T}{\partial P_{Gi}}\right) = 0 \qquad \text{or} \qquad \lambda = \eta_{LGi}\frac{dF_i}{dP_{Gi}} \tag{4.2}$$

where

$$\rho_i = \frac{\partial P_T}{\partial P_{Gi}}$$

is the incremental transmission losses which is a measure of the sensitivity of the system losses to an incremental change in the output of generating unit *i* when all other generating unit outputs are kept fixed and

$$\eta_{LGi} = \left(\frac{1.0}{1.0 - \rho_i}\right)$$

is the LaGrange penalty factor of generating unit *i*.
Equation (4.2) can be compactly written by inserting all the necessary value of penalty factor, incremental production cost as

$$\lambda = \frac{2P_{Gi}\gamma_i + \beta_i}{1.0 - \rho_i} \qquad \text{or} \qquad P_{Gi} = \frac{\lambda[1.0 - \rho_i] - \beta_i}{2\gamma_i} \tag{4.3}$$

From the equality constraint of equation (2.5) and putting the values of individual P_{Gi}s we have

$$\lambda = \frac{\left(2(P_T + \sum_{k=1}^{n_L} P_{Dk} + \sum_{i=1}^{n_G} P_{HDi}) + \sum_{i=1}^{n_G} \beta_i/\gamma_i \right)}{\sum_{i=1}^{n_G} (1.0 - \rho_i)/\gamma_i} \tag{4.4}$$

4.1.1 Approach

These equations described above are collectively known as the coordination equations, and the processes involved in the solution of this problem are described in the following steps:

Step 1: Initialization and actualization of all necessary power system data. Base case load flow is then carried out.

Step 2: Formation of bus admittance matrix and computation of impedance matrix by taking the inverse of the bus admittance matrix.

Step 3: Calculation of incremental transmission losses using the approximate equation as derived in [48]

$$\rho_j \approx 2 \cdot \sum_{k=1}^{n_B} \left(P_k a_{jk} - Q_k b_{jk} \right) \tag{4.5}$$

where

$$a_{jk} \equiv \frac{R_{jk} \cos(\delta_j - \delta_k)}{|V_j||V_k|} \quad , \quad b_{jk} \equiv \frac{R_{jk} \sin(\delta_j - \delta_k)}{|V_j||V_k|}$$

and R_{jk} is the real part of the element jk of the impedance matrix.

Step 4: Evaluation of system incremental cost λ^{μ} for iteration μ using the equation (4.4).

Step 5: Computation of individual generating unit power output from equation (4.3).

Step 6: Verification of the power balance condition from equation (2.11):

$$\left(\sum_{i=1}^{n_G} P_{Gi} - \sum_{k=1}^{n_L} P_{Dk} - P_T - \sum_{i=1}^{n_G} P_{HDi} \right) \le \varepsilon_P \quad ; \tag{4.6}$$

$\varepsilon_P \triangleq$ power balance tolerance.

If this condition is not satisfied, update the system λ by computing its new value $\lambda^{\mu+1}$ by means of extrapolation technique using the last two successive values of λ to estimate the next value [49]:

$$\lambda^{\mu+1} = \lambda^{\mu} + \frac{\sum_{k=1}^{n_L} P_{Di} + P_T + \sum_{i=1}^{n_G} P_{HDi} - \left(\sum_{i=1}^{n_G} P_{Gi}\right)^{\mu}}{\left(\sum_{i=1}^{n_G} P_{Gi}\right)^{\mu} - \left(\sum_{i=1}^{n_G} P_{Gi}\right)^{\mu-1}} \left(\lambda^{\mu} - \lambda^{\mu-1}\right) \tag{4.7}$$

where superscript (μ+1) indicates the next iteration being started, superscript (μ) is the iteration just completed, and (μ-1) denotes the immediately preceding iteration. Return to step 5 until the final convergence is achieved.

<u>Step 7:</u> Check if the calculated individual generating units' power output P_{Gi} are within the operating limits; if a power limit is subsequently found to be violated then set the generating unit output at corresponding limit $P_{Gi} = P_{Gi}^{min}$ or $P_{Gi} = P_{Gi}^{max}$ and eliminate the unit from the problem with the corresponding generation-load balanced constraint fixed at

$$\sum_{i=1}^{n_G} P_{Gi} = \sum_{k=1}^{n_L} P_{Dk} + P_T + \sum_{i=1}^{n_G} P_{HDi} - \sum_{m=1}^{n^+} P_{Gm}^{max} - \sum_{w=1}^{n^-} P_{Gw}^{min} \tag{4.8}$$

where

$n^- \triangleq$ number of units whose power output violate the lower limit

$n^+ \triangleq$ number of units whose power output violate the upper limit;

and return to step 3 and continue the calculations until final convergence is achieved.

<u>Step 8:</u> Specifying the optimal P_{Gi}s at their respective nodes and execute the final load flow to determine appropriate generation cost, total system losses and generating units' power schedule.

4.1.2 Realization

Specifically for the training simulator used in this work, all the required power system data of generating units, loads and network topology are retrieved from the process database by the corresponding program packet of **generation**

observer, **load observer**, and **topology evaluation**. An additional program packet was developed to retrieve the generating units' constant α, linear β and quadratic γ cost coefficients from the process database. The data are of the form outlined in chapter 3 (see table 3.2a). In addition to the data outlined in table 3.2a, the minimum loading conditions of various types of generating units are computed as a certain percentage of the rated power output using the information of table 3.2b.

After all the necessary power system data have been actualized, steps 1 to 8 described above are then applied to determine the optimal generating units power set-point schedule. An existing Newton-Raphson load flow program was used to carry out the load flow analysis. The power balance tolerance ε_P of equation (4.6) was specified at 0.0001% of total real power demand, and for the first iteration in step 6, the values of λ^0 and $\left(\sum_{i=1}^{n_G} P_{Gi}\right)^0$ were both set at zero.

After the final converged solution for both system λ and economic loading of the generating units has been found, a final load flow is then executed to compute:

- the appropriate generating units power set-point schedule,
- the total generation cost, $F_T = \sum_{i=1}^{n_G} F_i(P_{Gi})$ in CU/h, and
- the total system transmission losses P_T.

Initial and optimal generating units power set-point schedule and their identifiers in GDL format are then mapped into the optimal generating unit schedule list until they are required for further autonomous execution or for presentation to the operator to be adjusted manually.

4.2 Innovative GA-based Secure Real Power Dispatch

The solution of the economic load dispatch problem using the classical approach described above presents some limitations in its implementation. One of such limitations is that the lambda-iteration method assumes the cost coefficient to be a continuous function. The method breaks down when it is applied to a discontinuous function with prohibited zones or large steam turbine generating units which have a function of the form given by equation (2.8) discussed in chapter 2. Also, there is a high tendency for this approach to be caught at the local minima when the power system operating status is far outside the normal situation, as for instance during and after large disturbances.

For this reason, the method of GA was applied in this work to the active power dispatch problem to eliminate the limitations of the lambda-iteration enumerated above. In unit-based encoding, the number of bits constituting a chromosome increases with the number of units which may lead to some difficulties such as poor performance, long computational time and large memory demand for large-scale application. For this reason, a variation of the lambda-iteration approach in the form of a genetic algorithm was applied in this work to the real power dispatch problem using the lambda-based encoding. By encoding the normalized system incremental cost (lambda) within specified limits as determined by the minimum and maximum active powers of the actually synchronized units, see equation (4.11) instead of unit power output, the number of bits required for coding can be considerably reduced, since the number of bits of chromosome required is independent of the number of units. This feature therefore makes it attractive for large scale problems. Furthermore, additional operational constraints such as branch (line and transformer) apparent power flows limit violations can easily be incorporated into the objective function to eliminate branch overload problem. As will be shown later, this criterion is a dominating factor in the application implemented in this work.

4.2.1 Approaches

For either unit-based or lambda-based encoding GA real power dispatch, the description of the general procedure involved in their implementations is as follows:

4.2.1.1 Initialization

The GA real power dispatch starts with the choice of appropriate GA parameters and actualization of all the necessary power system data required for the computational process as described in chapter 3, section 3.6.2.

4.2.1.2 Encoding and Decoding Schemes

The encoding in which the problem is to be represented in the GA must be carefully designed to utilize the GA's ability to efficiently transfer information between chromosomes (strings) and the problem's objective function. Binary representation has been widely used for GA analysis because of the ease of binary number manipulation and the fact that GA theory is based on the binary alphabet. Two encoding mechanisms based on coding and decoding of equal system incremental production cost λ and generating units power output are considered; thus the names 'lambda based' and 'unit based' GA real power dispatch. These are described below:

4.2.1.2.1 Lambda-based Encoding and Decoding

In this approach, equal system incremental cost λ is used as coding parameter. The number of bits will be entirely independent of the number of generating units. The encoding parameter is the normalized system incremental cost λ^{nom} within the range 0 and 1. At the initialization stage, the initial chromosomes $C = [s_k^m]$, $m = 1,2,....n_p$; $k = 1,2,....b_l$, for n_p population size using b_l number of bits are randomly generated from the sets of uniform distribution ranging over 0 and 1, and $s_k^m \in [0,1]$.

Coding of n_p individuals is illustrated in figure 4.1 by using b_l=10 bits to code the normalized system incremental cost. The resolution of the solution depends on the number of bits used to represent the parameter. The more the encoding bits there are, the higher the resolutions and the slower the convergence.

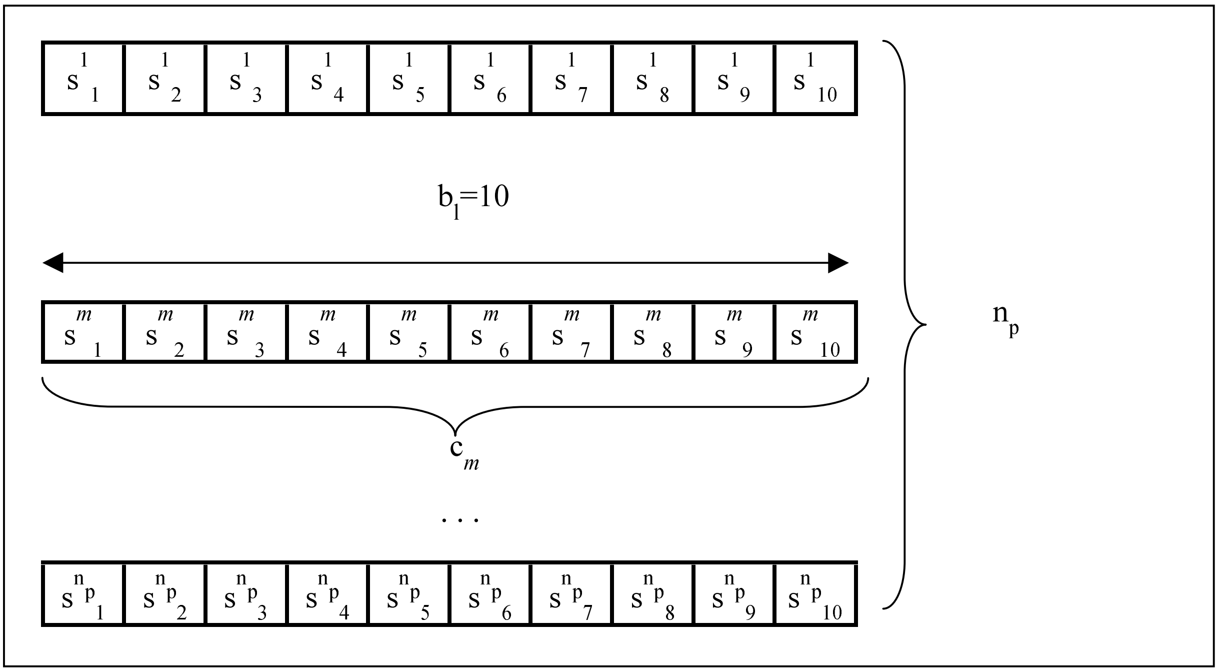

Figure 4.1 Encoding scheme of normalized system incremental cost

To evaluate the individual, the encoded strings are decoded to give the real value equivalent of the normalized system incremental cost. For the m^{th} individual of the population, the decoded value λ_m^{dec} can be computed using

$$\lambda_m^{dec} = \frac{\sum_{k=1}^{10} s_k^m \cdot 2^{b_l - k}}{2^{b_l} - 1} \tag{4.9}$$

The normalized system incremental cost is then transformed into actual system incremental cost by using

$$\lambda_{sys}^{m} = \lambda_{sys}^{min} + \lambda_{m}^{dec} \cdot (\lambda_{sys}^{max} - \lambda_{sys}^{min}) \tag{4.10}$$

where λ_{sys}^{min} and λ_{sys}^{max} are respectively the minimum and maximum values of system incremental cost determined from the actually synchronized units within their operating loading limits using the expressions

$$\begin{aligned} \lambda_{sys}^{min} &= \min\left(\frac{dF_i}{dP_{Gi}}\right) \\ & \qquad\qquad\qquad i = 1,2.....n_G \\ \lambda_{sys}^{max} &= \max\left(\frac{dF_i}{dP_{Gi}}\right) \end{aligned} \tag{4.11}$$

Applying the methods of the LaGrange and Kuhn-Tucker conditions [39,68] for n_G synchronized generating units, the problem is then reformulated for the m^{th} individual of the population as:

$$\begin{aligned} \lambda_{sys}^{m} &= \beta_i + 2\gamma_i P_{Gi} \quad \text{if} \quad P_{Gi}^{min} \le P_{Gi} \le P_{Gi}^{max} \\ \lambda_{sys}^{m} &\ge \beta_i + 2\gamma_i P_{Gi} \quad \text{if} \quad P_{Gi} = P_{Gi}^{max} \\ \lambda_{sys}^{m} &\le \beta_i + 2\gamma_i P_{Gi} \quad \text{if} \quad P_{Gi} = P_{Gi}^{min} \end{aligned} \tag{4.12}$$

subject to power balance constraint of equation (4.7). The cost coefficients β_i and γ_i are as defined by equation (2.11) and outlined in table 3.2a. Having determined the actual system incremental cost, the generation output of each unit can be evaluated from equation (4.12). If the computed generation output power is outside the operating limits, adjustment is made to the corresponding unit's power output to operate at the lower or upper limit depending on the situation.

This approach is illustrated with an example. Consider a system consisting of four thermal units in which the minimum and maximum values of system incremental cost λ_{sys}^{min} and λ_{sys}^{max} have been determined to be 8.38 and 12.61 CU/MWh respectively. For individual number 4 of the population, coded with b_l=15 bits in which the output of the random generator is 100011111101010, the decoded value is computed from equation (4.9) as

$$\lambda_4^{dec} = \frac{\sum_{k=1}^{15} s_k^4 \cdot 2^{15-k}}{2^{15}-1} = 0.56185$$

The actual system incremental cost can be determined using equation (4.10) as

$$\lambda_{sys}^4 = [8.38 + 0.56185(12.61 - 8.38)]\,\text{CU/MWh} = 10.76\,\text{CU/MWh}$$

Each generating unit's power output can be computed by substituting this value in equation (4.12).

4.2.1.2.2 Unit-based Encoding and Decoding

In this approach, each generating unit loading range is represented by a binary number. The number of bits required to represent each unit's output can be calculated after the resolution in unit output has been agreed upon. Consider for example a system consisting of n_G generating units each loaded within its limits $[P_{Gi}^{min}, P_{Gi}^{max}]$; the value P_{Gi} of unit i is represented by an approximate nearest integer bit length b_1^i given by

$$b_1^i = \log_2\left\{\frac{\left[\left(P_{Gi}^{max} - P_{Gi}^{min}\right) + P_\Delta\right]}{P_\Delta}\right\} \tag{4.13}$$

where P_Δ is the unit's power output resolution, and the total string length is obtained by concatenating the bits representing each unit. At the initialization stage, the chromosomes of n_G generating units for a population size n_p are randomly generated from the sets of uniform distribution ranging over the minimum and maximum power output.

An example of this representation is shown in figure 4.2 for a system with the number of generating units n_G=3, each coded with b_1^i=4 bits. To evaluate the unit power output levels, sub-strings of the chromosomes are extracted and decoded within the operating limits to give the real number equivalent values. The power output value of each generating unit i of the m^{th} individual in the population can be evaluated using the expression, see appendix D, equation (D.4)

$$P_{Gi}^m = P_{Gi}^{min} + \sum_{k=1}^{b_1^i} s_{km}^i \cdot 2^{b_1^i-k}\left\{\frac{P_{Gi}^{max} - P_{Gi}^{min}}{2^{b_1^i}-1}\right\} \tag{4.14}$$

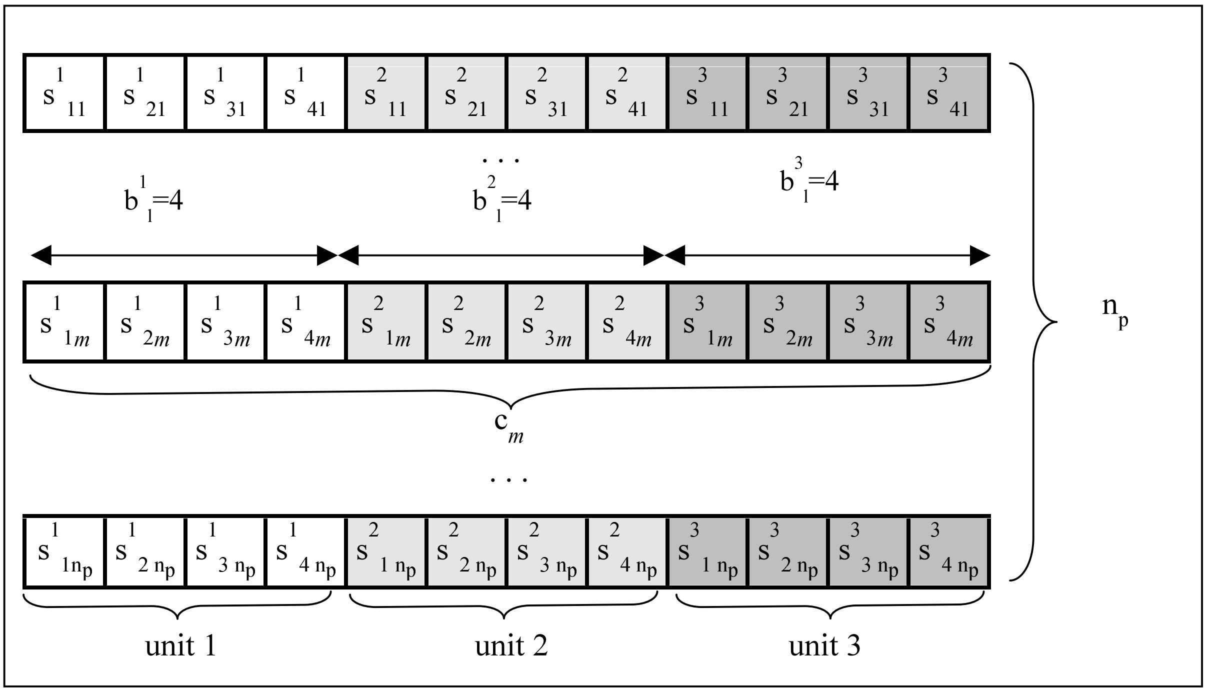

Figure 4.2 Encoding scheme of generating units

4.2.1.3 Function Evaluation and Treatment of Constraints

The evaluation process computes the fitness of each individual as a solution to the optimization problem. The objective function has two components: the fuel cost and penalty terms associated with system constraints violation. The system constraints to be satisfied are:

- power balance constraints of matching the sum of load demand, substation auxiliary power supply and system real power losses with that of generating units power output, see chapter 2, equation (2.11); and
- branch (transformer and line) apparent power flow limit, see chapter 2, equation (2.13).

Generating units' minimum and maximum loading constraints are taken care of in the encoding process as discussed above. Changes in each unit power output computed as control variable (see section 4.2.1.1 above) depending on the method, i.e. lambda based or unit based, are mapped into the data of their respective power units. The power flow program is then actuated, delivering the branches' (lines and transformers) apparent power flows and losses associated with each individual schedule, and the fitness of each individual constituting the population is then computed.

The rank-based reproduction scheme proposed by [69] was finally adopted in this work. In this scheme, the function to be optimized is redefined to be less than 1.0, and the constraints are mapped into a range greater than 1.0. This ensures that all the feasible points are always preferred while at the same time the function can be normalized in order to ameliorate the scale difference between the ranges of constraints, and to allow the infeasible points to compete for reproduction. The load dispatch problem taking into consideration n_{Br} branches' apparent power flow limit as well as power balance constraints is then transformed into the fitness objective function and can be computed for the m^{th} individual of the population using

$$F_{fit}^{m} = 1.0 + \eta'_{GA} \cdot K\left(\varphi + \sum_{l=1}^{n_{Br}} (|S_l| - S_l^{max})\right) \quad ; \forall \quad |S_l| > S_l^{max} \tag{4.15}$$

where

$m=1,2,.....n_p$ and $K = \dfrac{1.0}{S_{Base}}$

η'_{GA} is the penalty term the value of which is chosen in such a way that the objective function is scaled below the minimum penalty value; it is assigned a value of 1.0. K is a scaling constant expressed as a fraction of the base power value S_{Base} (see appendix A).

Conventional GAs are formulated as maximization problem. Since the real power dispatch problem is a minimization problem, it is transformed into a maximization problem using

$$\min\left[F_{fit}^{m}\right] \Leftrightarrow \max\left[\frac{1.0}{F_{fit}^{m}}\right] \tag{4.16}$$

The individual fitness is calculated from the fitness objective function of equations (4.15) and (4.16) using the expression of the form:

$$f_m^{ind} = \frac{1.0}{\left\{1.0 + \eta'_{GA} \cdot K\left[\varphi + \sum_{l=1}^{n_{Br}} (|S_l| - S_l^{max})\right]\right\}} \quad ; \forall \quad |S_l| > S_l^{max} \tag{4.17}$$

After computing the fitness of each individual, the parents then undergo the genetic operation of selection and crossover; each pair creates a child having some mix of the two parents. The process of selecting and mating of individuals continues until a new generation is reproduced. The chromosome of each individual constituting the population is subjected to mutation. The elite preserving strategy is also applied. Subsequently, the fitness of the individuals of the new generation is evaluated, and this procedure continues until the convergence criterion is reached. The algorithm is defined to converge when the following conditions are satisfied:

- There are no branch apparent power flow limit violations **AND**
- the power balance constraint given by equation (4.6) is less than a certain tolerance ε_p **AND**
- if the average fitness of the population exceeds some fraction of the best fit in the population (the value used here is 0.999) **OR**
- there is no improvement in the incumbent solution after a specified number of generations (typical value of 5-50 generations).

If the above conditions are not met, the algorithm automatically stops at the pre-defined maximum number of generations.

After the algorithm has converged, the most fit individual of this generation is chosen as optimum solution to the problem. The optimal generation schedule in form of assigned P_{Gi}s at the respective generating units *i* is specified, and the final load flow is executed to determine:

- an appropriate generation schedule,
- the total generation cost and
- the total system real power losses.

The above described general procedure for both methods is clearly depicted in figure 4.3, and details of processes involved in the computations of the individual fitness for lambda based encoding GA and unit based encoding GA are respectively shown in figures 4.4 and 4.5.

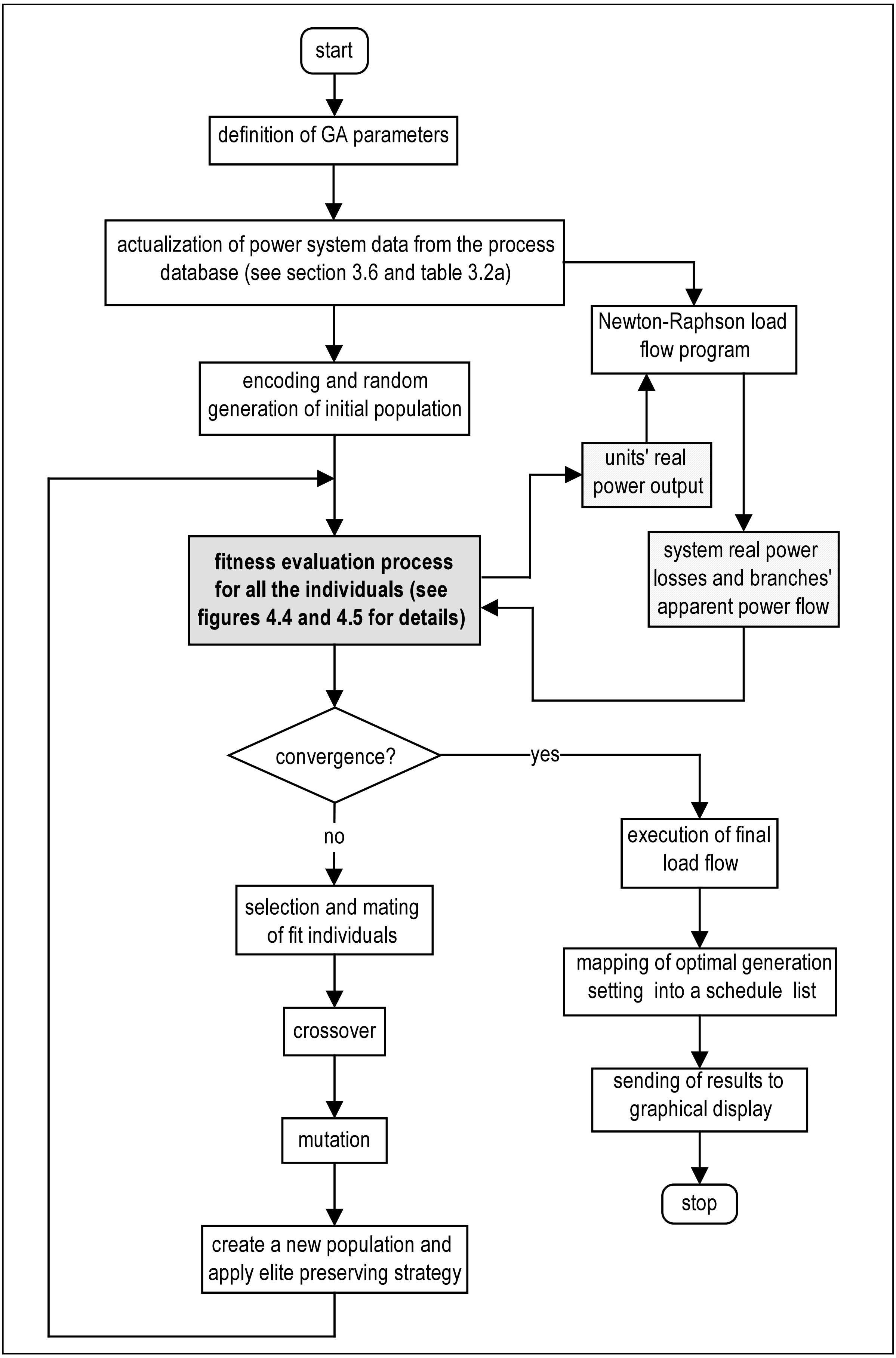

Figure 4.3 Flow chart of GA based real power dispatch strategy

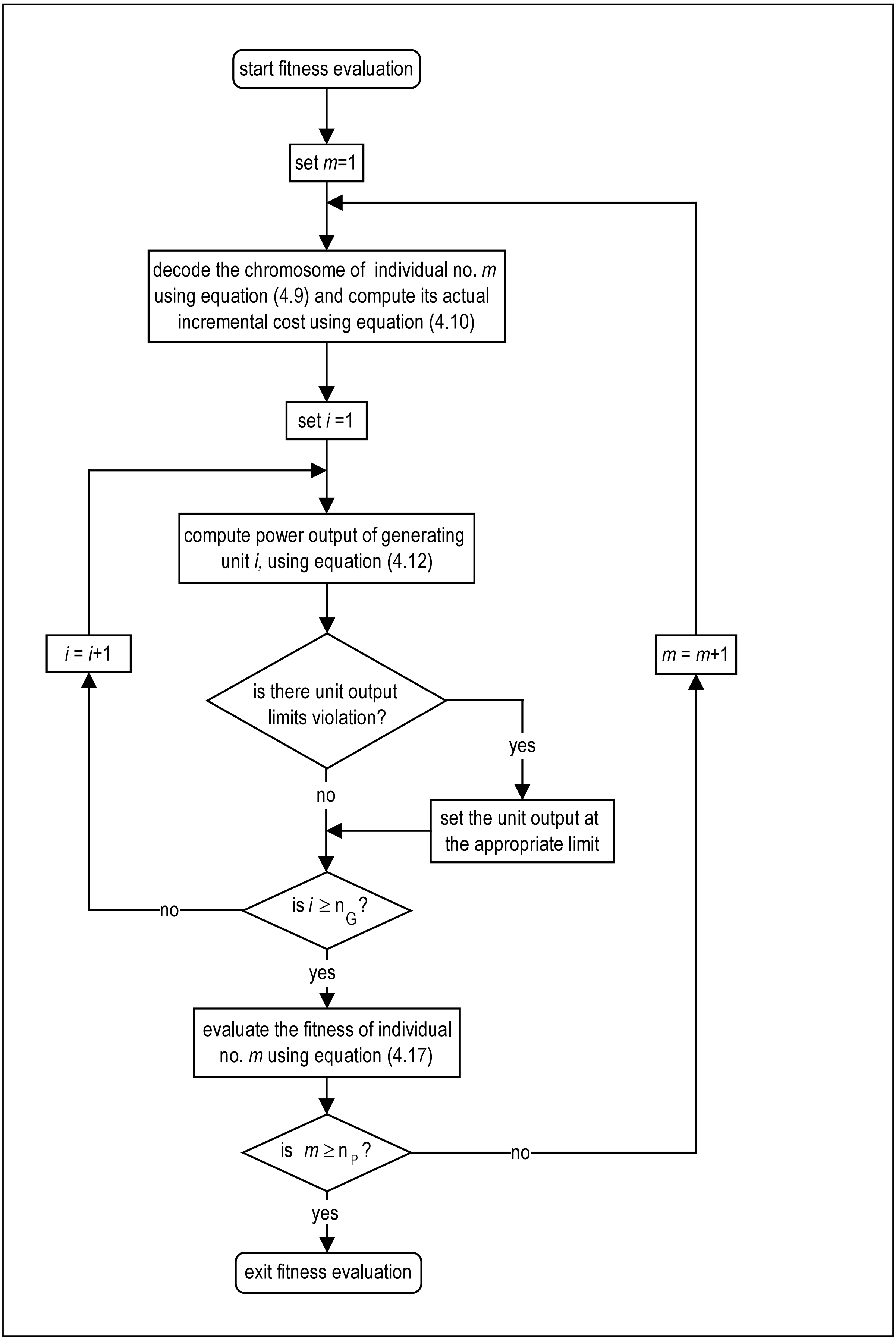

Figure 4.4 Lambda-based GA fitness evaluation process (as described in section 4.2.1.2.1)

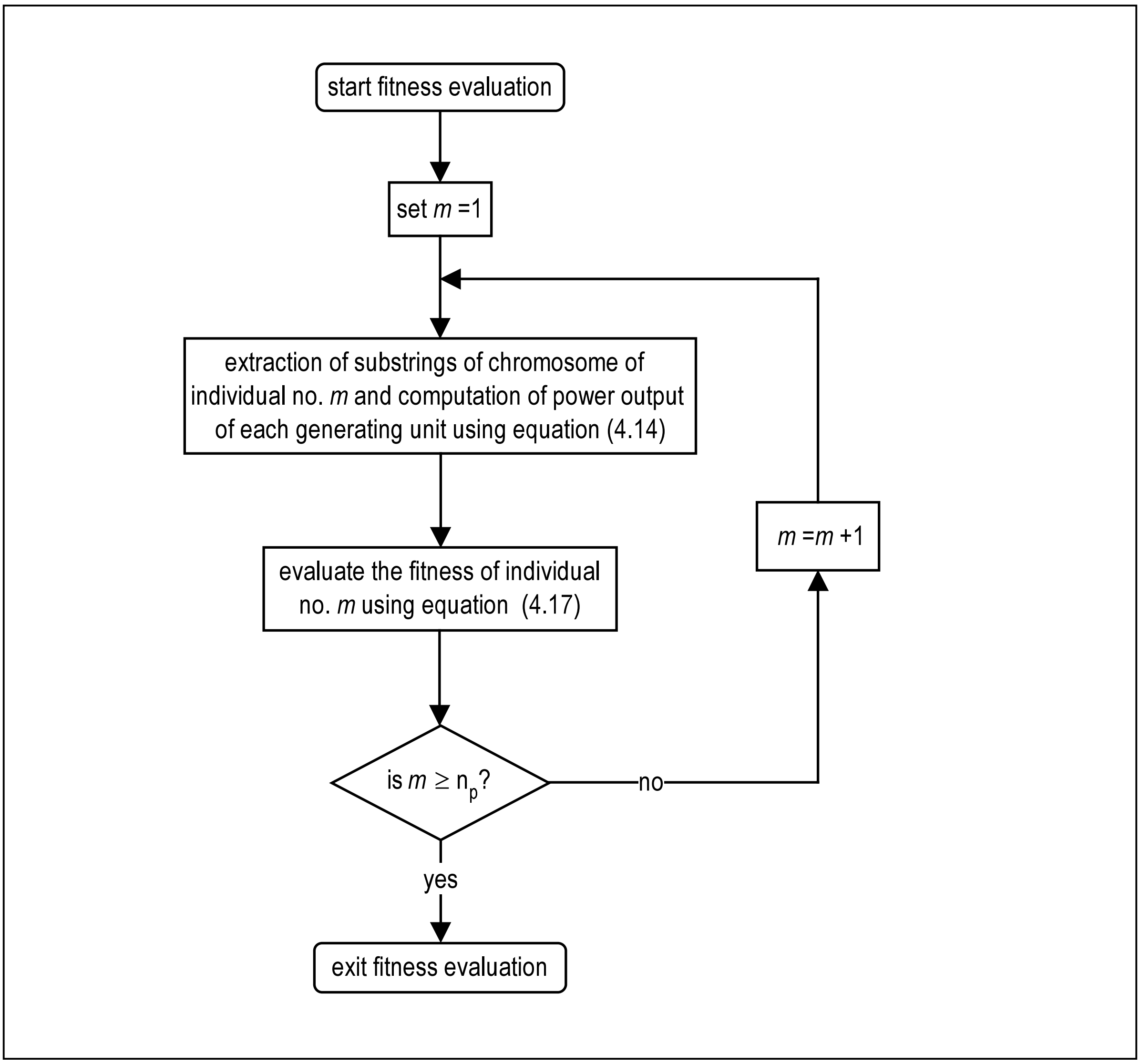

Figure 4.5 Unit-based GA fitness evaluation process (as described in section 4.2.1.2.2)

4.2.2 Realization

The processes described in section 4.2.1 were realized for both lambda and unit based encoding schemes of the GA based economic load dispatch. At the initialization stage, basic control parameters are defined: Population size (n_p), encoded parameter resolution, number of children per pair of parents, elite preserving strategy and genetic operation probabilities such as crossover rate (σ_c) per pair of parents, mutation rate (σ_m) per bit and creep mutation rate (σ_{cr}) per parameter. The values used in this work are as depicted in table 4.1. Furthermore, all the required power system data must be actualized. Specifically for the training simulator used in this work as a substitute for a power system control, the data of generating units, loads and network topology are retrieved from the process database by the corresponding program packet of **generation observer**, **load observer**, and **topology evaluation** as outlined in chapter 3, table 3.2a. An additional program packet was developed to retrieve the generating units' constant α, linear β and quadratic γ cost coefficients from the process database. In addition to these data, the minimum loading conditions of various types of generating units are computed as a certain percentage of the rated power output using the information of table 3.2b.

At the start of the evolution process, an initial population is randomly created within the parameters' minimum and maximum values by using a random generator. The fitness of each individual of the population is evaluated according to figures 4.4 and 4.5. The convergence criteria outlined above are tested. The power balance tolerance ε_P was specified at 0.0001% of total real power demand, and the improvement in generation convergence was set at 20 generations. If the convergence criteria are satisfied, the algorithm is exited, otherwise the genetic operations of selection, crossover and mutation are applied.

Tournament selection is used to select mates according to their fitness. The parents then undergo uniform crossover; each pair creates a child having some mix of the two parents. The process of selecting randomly pairs and mating the stronger individuals is continued until a new generation is reproduced.

Table 4.1 Applied parameters for GA based real power dispatch

Parameter	Value
Population size (n_p) individuals	20
Mutation rate (σ_m) per bit	σ_m=1.75/$l_c \cdot n_p$ *
Creep mutation rate (σ_{cr}) per parameter	0.04
Uniform crossover rate (σ_c) per pair of parents	0.5
Elite preserving strategy?	yes
Maximum number of generations (gen^{max})	100
Improvement in generation convergence	20
Parameter resolutions	for lambda-based encoding GA, (b_l=15 bits) for unit-based encoding GA (b_l^i=20 bits, i=1,2,...n_G)
Number of offspring per pair of parents	1

* l_c is the chromosome length ; n_p is the population size

The chromosome of each individual constituting the population is subjected to mutation and creep mutation. During the evolution process, the elite preserving strategy is applied by checking if the fittest individual of the last population is reproduced in the current generation; if not, a randomly selected individual is replaced by the old elite member. From the initial generation, a new population of the same size is generated using the genetic operations. Subsequently, the fitness of the individuals of the new generation are evaluated, and this procedure continues until the convergence is reached.

After the algorithm has converged, the most fit individual of this generation is chosen as optimum solution to the problem. Specifying these optimal generation schedule P_{Gi}s at their respective generating units, a final load flow is executed to compute:

- appropriate generation schedule,
- total generation cost $F_T = \sum_{i=1}^{n_G} F_i(P_{Gi})$ in CU/h, and
- total system real power losses.

Initial and optimal generation set-points scheduled, the generating units' identifiers in GDL format and their total number are then mapped into the optimal generating unit schedule result file. This system object file of generation control variables can be retrieved and passed on to the super-ordinate expert system for further autonomous execution, or for presentation to the operator to be adjusted manually as discussed in chapter 6. The results of the GA evolution process in the form of convergence behavior and production cost reduction are sent to a program for graphical display.

4.3 Simulation Results

Both of the above described and implemented GA approaches as well as the LaGrange approach were tested on two different real power systems: Duisburg 110/25/10 kV municipal and a part of the German high voltage 400/230/110 kV transmission system, both of them replicated on the operator training simulator in full operational detail [20] and briefly described in chapter 7. A multitude of test cases were performed on both networks, and samples of typical simulation results obtained for the above described three approaches are presented below:

4.3.1 Illustrative Examples with Duisburg Power System

There are 5 power units: 4 thermal units and a gas turbine unit. The overview diagram of this system is presented in chapter 7 (see figure 7.1). The system consists of 95 branches (lines / transformers), and contingency cases can be considered by outage of any of the branches. Realistic generating units' cost coefficients given as quadratic curves were obtained from [24,34,39] and are presented in table A.1 (see appendix A).

Scenario 1

The system was assumed to be operating in normal state with all loads supplied. For a total system demand of 249.0 MW, the optimal power dispatch of each of the four operating thermal generating units using the above described three methods of classical economic load dispatch, unit based encoding GA and lambda based encoding GA real power dispatch is as shown in figure 4.6. Each generating unit is named in GDL format reflecting the operators' terminology as described in chapter 3.

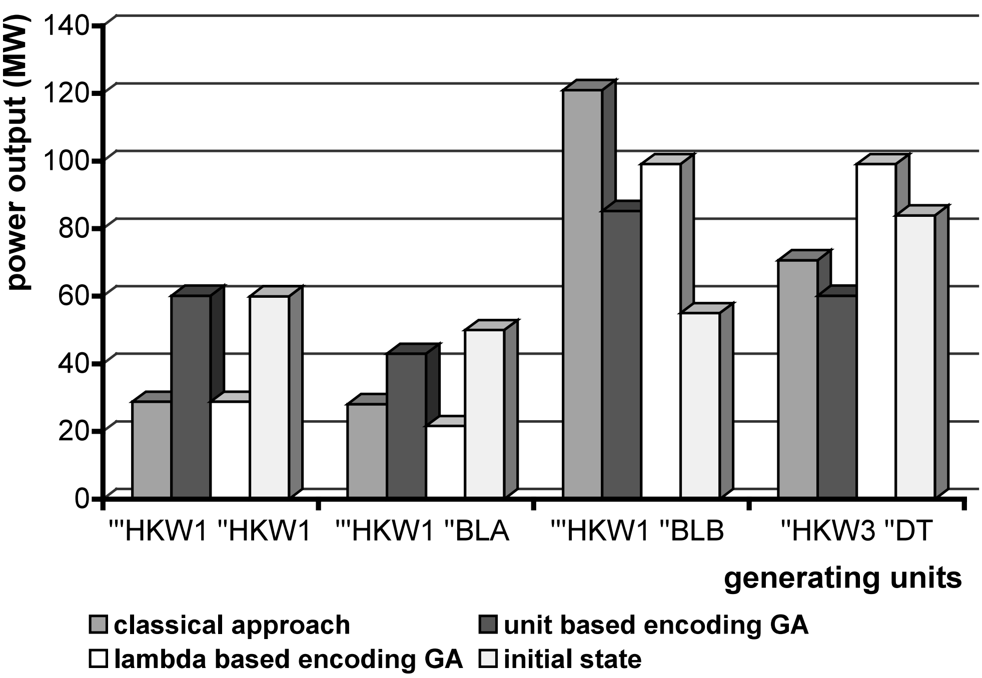

Figure 4.6 Re-dispatch of generating units for Duisburg municipal power system (scenario1)

From table 4.2, it can be seen that the three methods were able to solve the economic load dispatch problem, and that the lambda based encoding GA procures the least production cost compared with the other methods, although connected with highest total system real losses. The classical approach has a lower processing time compared with both GA methods.

Table 4.2 Results and comparison of three approaches for Duisburg power system (scenario 1)

Results	**Approaches**		
	Classical approach	**Unit based encoding GA**	**Lambda based encoding GA**
Initial production cost (CU/h)	2497.94	2497.94	2497.94
Final production cost (CU/h)	2432.08	2483.21	2424.68
Total power generation (MW)	248.23	248.21	248.61
Initial total real power losses (MW)	3.05	3.05	3.05
Final total real power losses (MW)	3.04	3.00	3.11
Generations taken to converge		19	41
Processing time (seconds)**	27	85*	180*

CU is fictional currency unit ; *time taken to fulfil the convergence criteria
**measured on 25 MHz HP-Apollo workstation

The evolution behavior of both GA approaches within 100 generations is also depicted in figures 4.7 and 4.8; to compare the convergence behavior of both GA approaches applied to the same sample scenario, maximal and average fitness values of individuals within generations according to equation 4.17 are given. It took unit based encoding GA approach 19 generations (85 seconds) to fulfil the convergence criteria defined in section 4.2.1.3 while lambda based encoding GA approach needed 41 generations (180 seconds) to converge. Unit based encoding GA thus converges faster in fewer generations. Lambda based encoding GA however needs a lower time (416 seconds) to process 100 generations than unit based encoding GA (423 seconds) since its chromosome length is shorter.

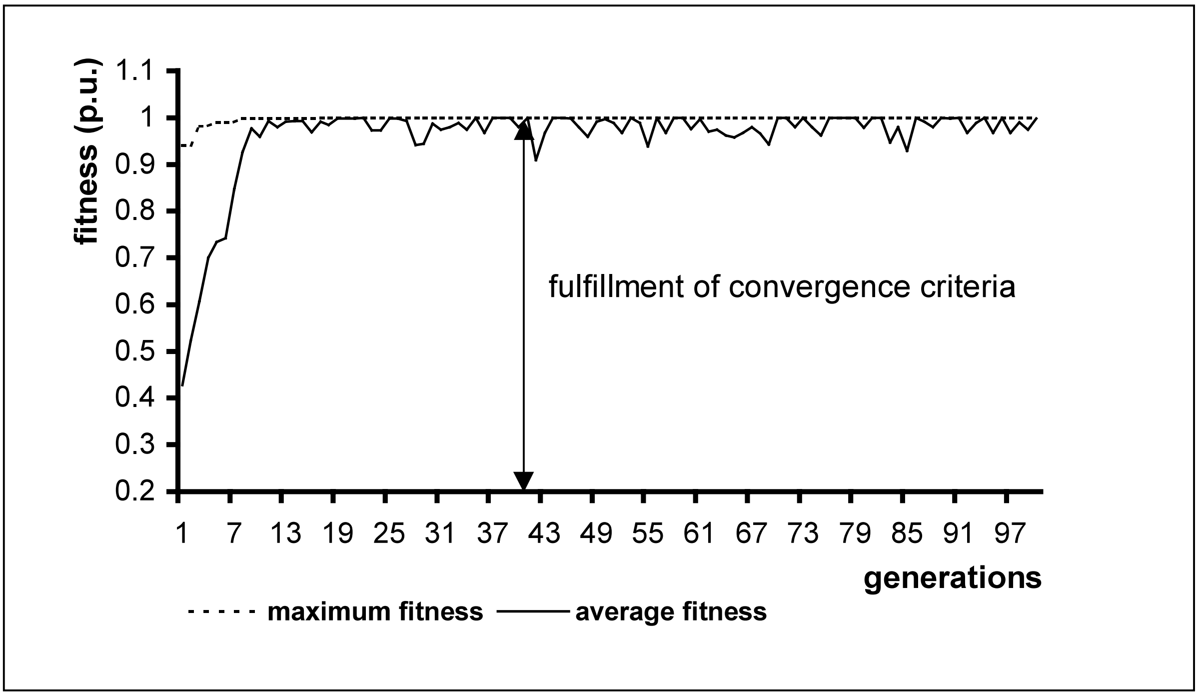

Figure 4.7 Convergence characteristics of lambda based encoding GA for Duisburg power system (scenario 1)

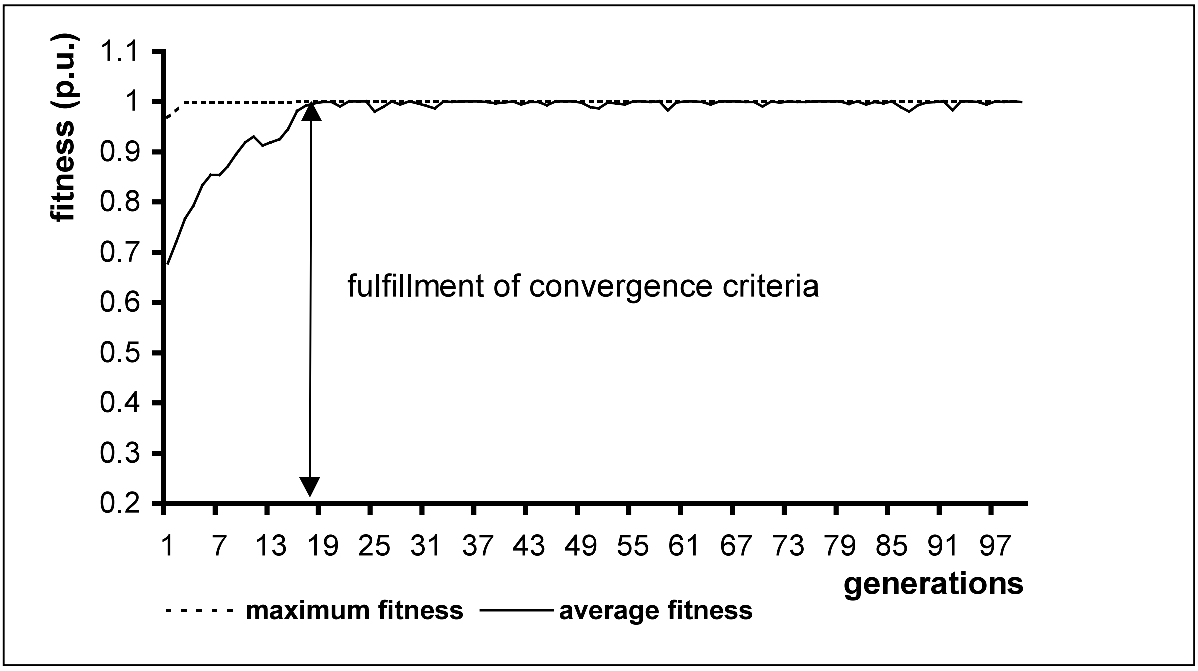

Figure 4.8 Convergence characteristics of unit based encoding GA for Duisburg power system (scenario 1)

Scenario 2

Here a scenario was created from the operating state of the network considered in scenario 1 which led to the violation of the apparent power flow limit of transmission line connecting substations HKW1 and OST1 (S^{act}=78.2 MVA; S^{max}=58 MVA i.e. 35% overload). The three approaches of classical economic load dispatch, unit based encoding GA and lambda based encoding GA were applied to re-dispatch the generation in order to eliminate this overload problem while at the same time operating at optimum state. Secured optimal real power dispatch of each of the four on-line thermal generating units using the three approaches are presented in figure 4.9 showing the adjustment to be made from the initial status.

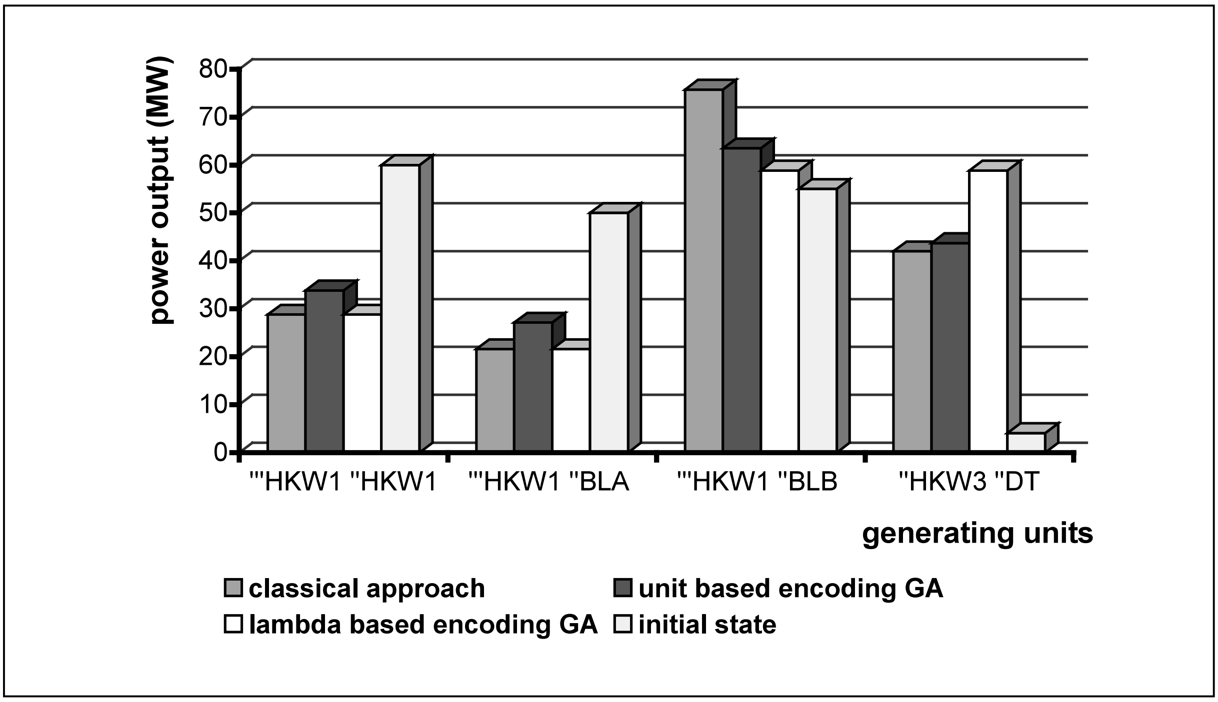

Figure 4.9 Re-dispatch of generating units (disturbance scenario of Duisburg power system)

Table 4.3 also depicts results of the economic load dispatch comparing the three methods. All three methods were able to eliminate the overloads problem while fulfilling the objective of minimizing the generation cost after the suggested optimal dispatch has been assigned to each individual generating unit. The lambda based encoding GA has the lowest total generation cost compared with the other methods. Both GA approaches are inferior to the classical approach in terms of processing time.

Table 4.3 Summary of results and comparison of three approaches for Duisburg power system (disturbance scenario)

Results	Approaches		
	Classical approach	Unit based encoding GA	Lambda based encoding GA
Initial production cost (CU/h)	1828.74	1828.74	1828.74
Final production cost (CU/h)	1724.81	1731.48	1721.68
Total power generation (MW)	168.14	168.12	168.09
Initial total real power losses (MW)	3.41	3.41	3.41
Final total real power losses (MW)	3.03	3.01	2.97
Generation convergence		16	17
Processing time (seconds)**	26	73*	76*

CU is fictional currency unit ; *time taken to fulfil the convergence criteria
**measured on 25 MHz HP-Apollo workstation

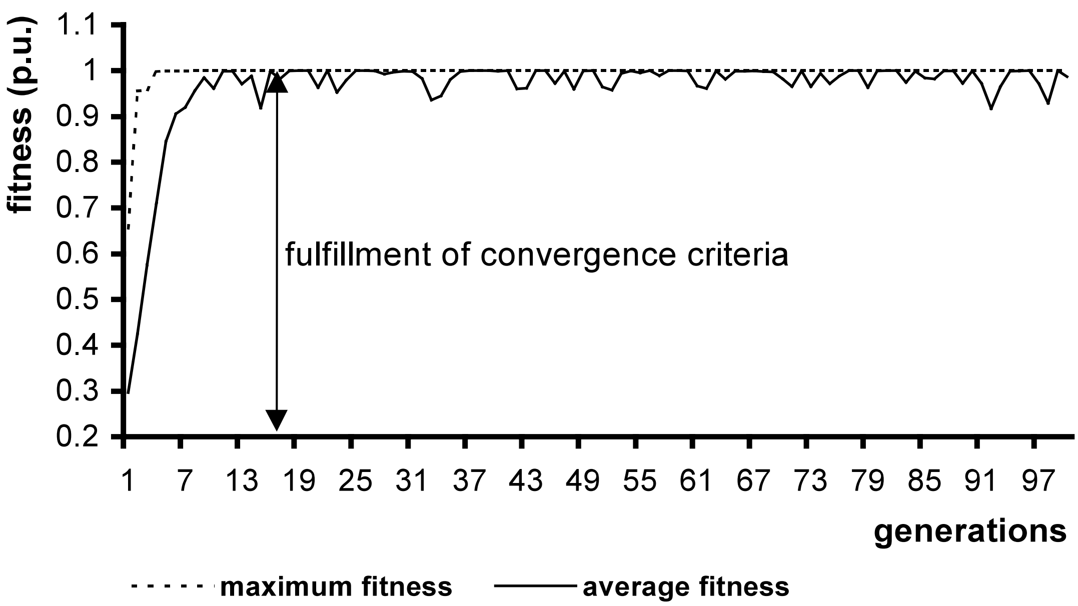

Figure 4.10 Convergence characteristics of lambda based encoding GA for Duisburg power system (disturbance scenario)

The evolution behavior of both GA approaches within 100 generations is depicted again in figures 4.10 and 4.11. The unit based encoding GA also satisfies all the convergence criteria in fewer generations than the lambda based encoding GA approach and has a more stable convergence behavior. It took lambda based encoding GA 412 seconds to process 100 generations while unit based encoding GA needed 425 seconds.

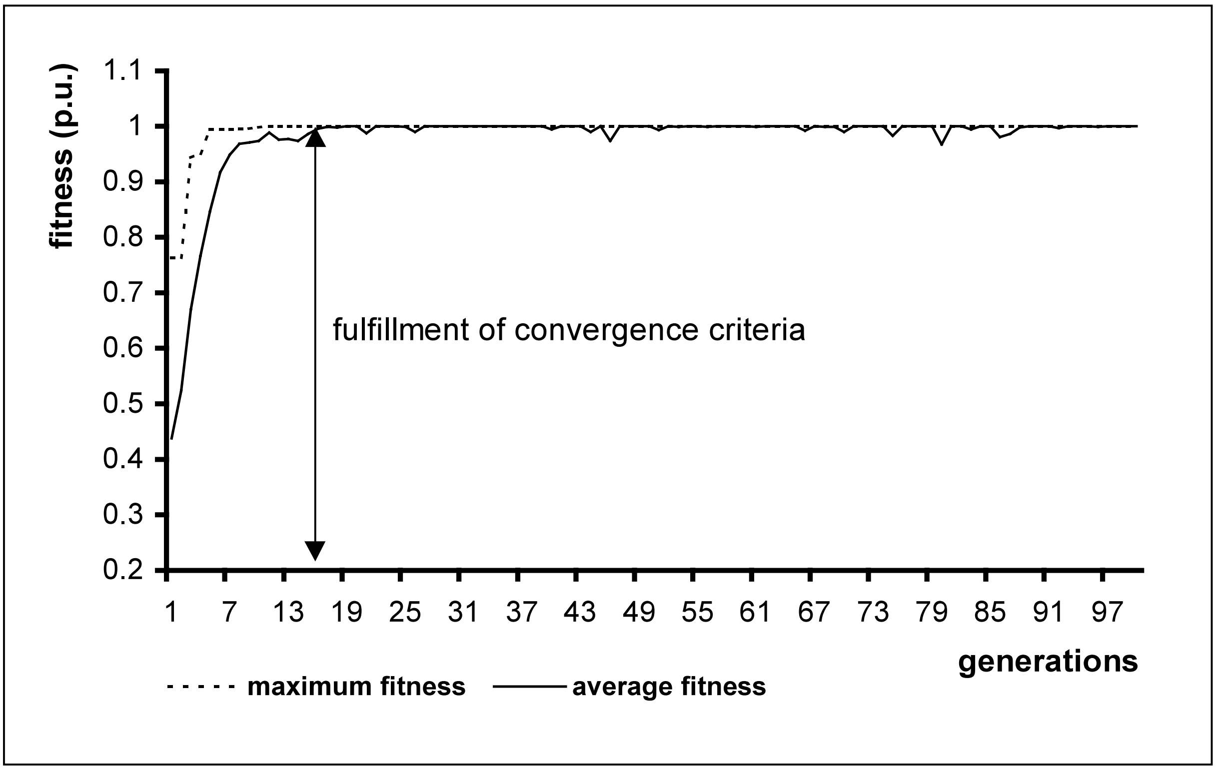

Figure 4.11 Convergence characteristics of unit based encoding GA for Duisburg power system (disturbance scenario)

4.3.2 Illustrative Example with 400/230/110 kV Transmission System

The real power dispatch approaches were also applied to a larger transmission network in order to demonstrate their capability and performance. The system under regard consists of 23 generating units. The overview diagram of this system is as presented in figure 7.3, chapter 7. Realistic generating units cost coefficients given as quadratic curves were obtained from [24,34,39], and operating limits are as presented in table A.2 (see appendix). For the scenario investigated here, there are 13 generating units actually synchronized, 10 of which are thermal units, two gas turbine units and one pressurized water reactor unit.

Optimal power dispatch of each of the 13 generating units using the above described three methods of classical economic load dispatch, unit based encoding GA and lambda based encoding GA real power dispatch is as shown in figure 4.12.

In table 4.4, results of the economic load dispatch are also presented comparing the three methods. It can be seen from table 4.4 that the three methods were able to solve the economic problem, but here the classical approach has the least production cost compared with GA approaches, and lambda based encoding GA is close in accuracy to the classical approach. Processing time required by both GA approaches are higher compared with that of the classical approach.

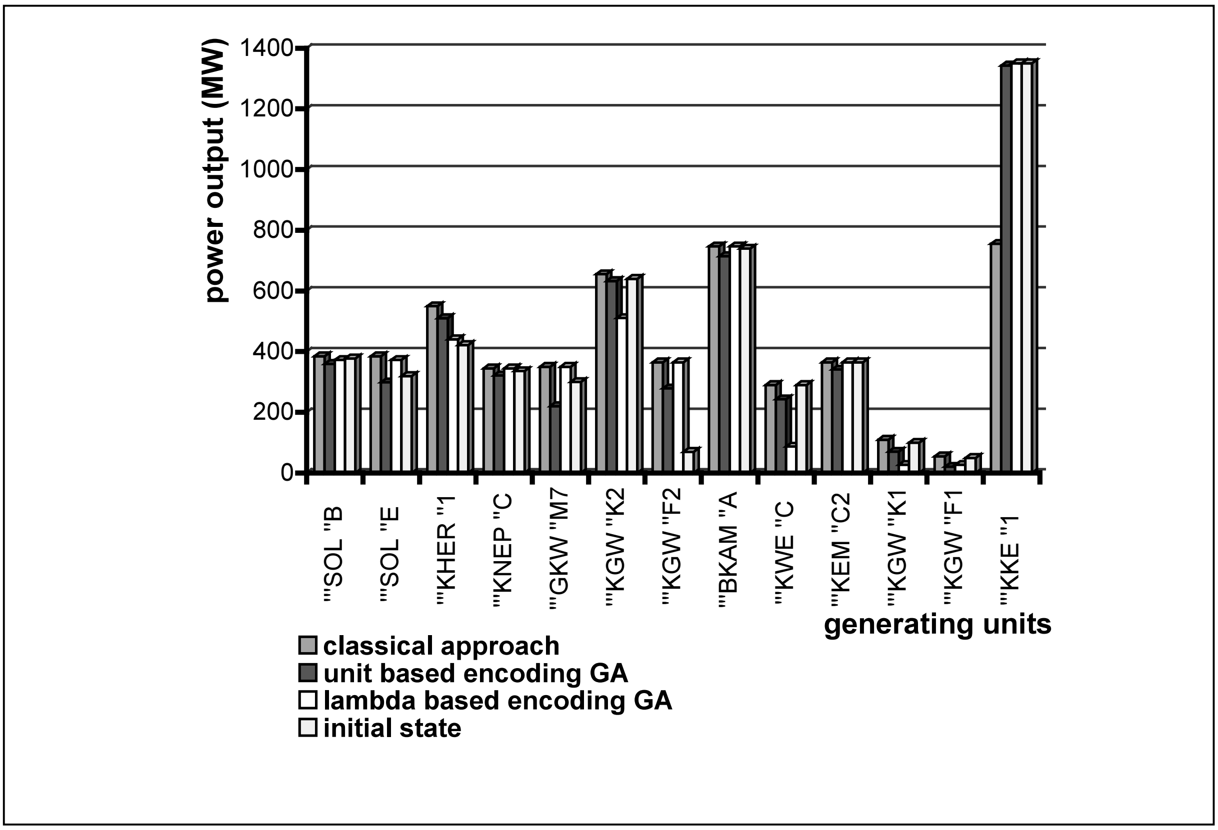

Figure 4.12 Re-dispatch of generating units (high voltage transmission system)

Evolution characteristics of both GA approaches in 100 generations are equally depicted in figures 4.13 and 4.14. From both figures and table 4.4, it can be noted that unit based encoding GA also satisfied the convergence criteria in fewer generations than lambda based encoding GA approach. The processing time required by unit based encoding GA is 542 seconds while lambda based encoding GA approach needs 839 seconds. Conversely, Lambda based encoding GA required a lower processing time to process 100 generations (1484 seconds) compared with its unit based encoding GA (1547 seconds) counterpart since its chromosome length is shorter.

Table 4.4 Summary of results and comparison of three approaches (high voltage transmission system)

Results	Approaches		
	Classical approach	Unit based encoding GA	Lambda based encoding GA
Initial production cost (CU/h)	50679.91	50679.91	50679.91
Final production cost (CU/h)	49666.70	50356.45	50046.41
Total power generation (MW)	5357.40	5356.45	5359.60
Initial total real power losses (MW)	52.87	52.87	52.87
Final total real power losses (MW)	52.31	51.36	54.53
Generations taken to converge		35	55
Processing time (seconds)**	69.8	542*	839*

CU is fictional currency unit *time taken to fulfil the convergence criteria
**measured on 25 MHz HP-Apollo workstation

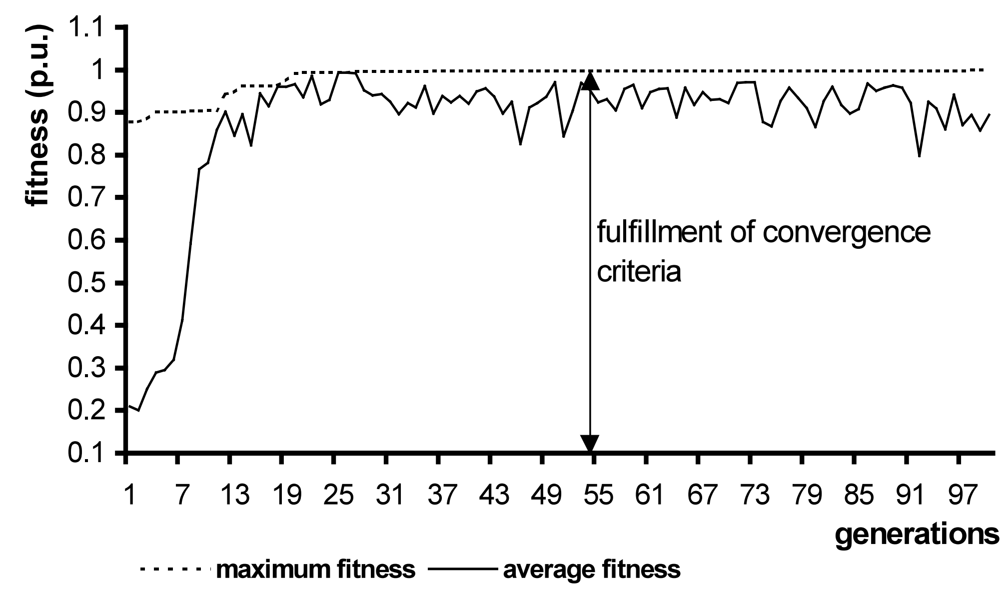

Figure 4.13 Convergence characteristics of lambda based encoding GA for high voltage transmission system

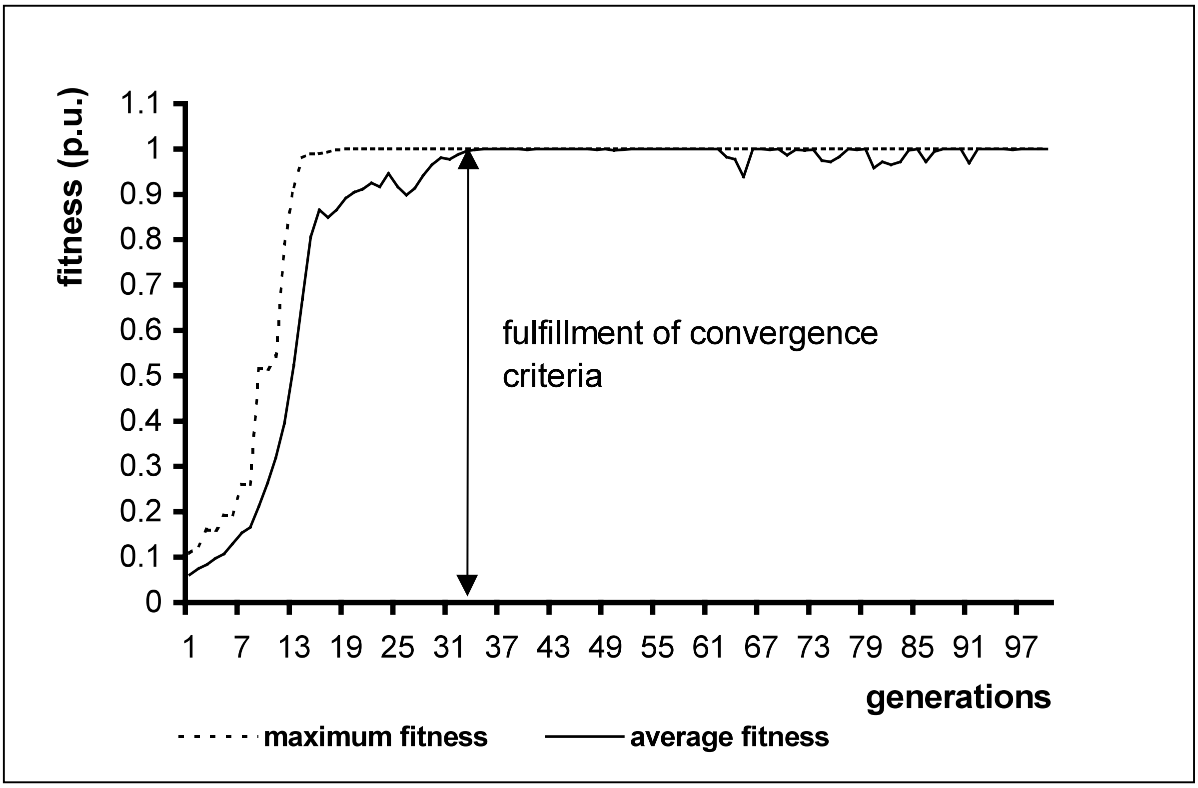

Figure 4.14 Convergence characteristics of unit based encoding GA for high voltage transmission system

4.3.3 Behavior of Unit based Encoding GA with the Number of Bits

In order to determine the optimal number of bits required to code each of the generating unit, a multitude of tests was performed on the Duisburg municipal network of scenario 2 with the number of bits varying between 10 and 35. From table 4.5, it can be seen that the optimal number of bits required to code each unit is 20 because it procures a good production cost and converges in fewer generations. Also, beyond 25 bits the performance became poorer, and when the bits per generating unit were increased to 35 (not shown in the table), the unit based encoding GA was even unable to solve the problem.

Table 4.5 Summary of results showing dependency of unit based encoding GA on the number of bits

Results	Number of bits per generating unit				
	10	**15**	**20**	**25**	**30**
Convergence time (seconds)*	259	78	70	127	110
Generation taken to converge	60	17	15	28	24
Initial production cost (CU/hr)	1829.14	1829.14	1829.14	1829.14	1829.14
Final production cost (CU/hr)	1734.22	1732.29	1732.01	1731.99	1738.46
Initial total real power losses (MW)	3.48	3.48	3.48	3.48	3.48
Final total real power losses (MW)	3.06	3.05	3.08	3.05	3.08
Total power generated (MW)	168.16	168.15	168.08	168.15	168.18

CU is fictional currency unit *measured on 25 MHz HP-Apollo workstations

4.4 Concluding Remarks

Simulation studies conducted on both municipal and transmission networks revealed that all the above approaches can solve the problem of economic load dispatch. Due to the fact that branch apparent power flow limits constraint was easily incorporated into the fitness function of the GA solution, a genetic computation approach was preferred over the classical approach where additional rigorous computational methods of generation shift distribution factors and generalized generation distribution factors [43,55] are involved in taking care of the branch constraints. Under this perspective, the higher computational time of the GA approaches developed in this work could be compensated. Furthermore, use of modern computer hardware could considerably contribute to alleviate the computational time problem.

A comparative analysis of the results of both GA approaches shows that the lambda based encoding GA is more accurate in terms of generation cost than the unit based encoding GA, but needs more generations to converge. For large scale application, the chromosome lengths required by unit based encoding GA increase proportionally with the number of units. This thus leads to high computational time and poor performance of the GA. In view of these facts, the lambda based encoding GA was intended to be finally dedicated as a sub-function of a hybrid system of state assessment and enhancement as described in chapter 6, procuring for the removal of branch overloads to normal operating conditions from states where the system is far outside optimal operation.