# Evolutionary Programming Based Method for Evaluation of Power Flow

**Kit Po Wong, Jason Yuryevich and An Li**
A.I. and Power Systems research Group
Electrical Engineering Dept.
University of Western Australia
Nedlands, Australia, 6907

## Abstract

This paper develops an Evolutionary Programming based load-flow algorithm. A number of solution acceleration techniques are developed to enhance the performance of the basic EP algorithm. Power system specific information is also incorporated into the EP algorithm to speed its convergence. Finally the algorithm is enhanced through the addition of the Jacobian matrix into the solution process which provides a very powerful and robust load-flow method. The developed method is validated through application on a number of IEEE test systems.

## 1 INTRODUCTION

Evolutionary computation techniques are becoming an accepted solution technique in many areas of problem solving. One field where it has been applied to many problems is that of power systems engineering [1, 2, 3]. In these fields, evolutionary computation brings with it the advantage of being able to solve more detailed problems where present and more classical techniques fail to produce acceptable solutions.

The load-flow problem [4, 5] is of critical importance in the day to day operation of modern power systems. By far the most popular method of solution for the load-flow is that of the Newton-Raphson load-flow (NRLF). The NRLF method has the advantage of being fast, converging in usually fewer than 6 iterations. However, if the system is heavily loaded or stressed the NRLF may fail to converge. In practice, it is in these situations when the loadflow solution is of great interest. The problem itself is one of very high dimensions with typical power systems having the order of thousands of nodes within them. This presents a large and highly constrained problem with practical significance.

Evolutionary programming (EP) [6, 7] is a stochastic optimisation technique based on the mechanics of evolution. It differs from classical calculus based methods in that it works with a population of solutions rather than a single solution. Through the operators of mutation and selection, this population is evolved towards the global optimum solution. Due to the nature of EP, it performs a parallel search of the solution space for optimal solutions. This parallelism can prevent premature convergence to local optima as well as providing alternative search paths if individual solutions become infeasible, unlike pure gradient based approaches.

The basic EP algorithm does not require any gradient information. However when the dimensions of the problem become large as in the case of loadflow, evolutionary computation methods can take an unacceptable period of time to converge or fail to converge completely. To overcome this limitation the gradient may be used to accelerate the population towards the global optimum. Inclusion of gradient information greatly enhances the performance of the EP, but the EP itself remains as the global optimisation routine with the gradient used as a local search about certain candidates.

## 2 THE LOADFLOW PROBLEM

The objective of the loadflow problem is to determine the voltages and power flows throughout a power network for a given operating condition. This problem may be formulated mathematically as follows. Let there be a total of $N$ nodes in the power system. If node $i$ is a load (PQ) node where the active and reactive power loads are known, the mismatch equations at that node are:

$$\begin{aligned} \Delta P_i &= |P_i^{sp} - P_i| \\ \Delta Q_i &= |Q_i^{sp} - Q_i| \end{aligned} \tag{1}$$

where $\Delta P_i$, $\Delta Q_i$ are the active and reactive power mismatches of node $i$ , $P_i^{sp}$, $Q_i^{sp}$ are the specified active and reactive powers for node $i$ and $P_i$, $Q_i$ are the calculated active and reactive powers for node $i$.

In the case of a generation (PV) node $i$ the voltage magnitude $|V_i|$ is specified as well as the active power generation $P_i^{sp}$. $Q_i$ is an unknown quantity to be determined at the solution. So in the case of a generation node the mismatch equations are:

$$\begin{aligned} \Delta P_i &= |P_i^{sp} - P_i| \\ \Delta V_i &= |V_i^{sp} - V_i| \end{aligned} \tag{2}$$

where $\Delta V_i$ is the voltage mismatch at node $i$, $V_i^{sp}$ is the specified voltage at node $i$ and $V_i$ is the present voltage at node $i$. In the present formulation, the voltage of generator nodes is always forced to be the specified value, hence $\Delta V_i$ is zero. Summing over every node in the system we can form the total squared mismatch of the system:

$$H = \sum_{i \epsilon N_{pq}+N_{pv}} \Delta P_i^2 + \sum_{i \epsilon N_{pq}} \Delta Q_i^2 \tag{3}$$

where $N_{pq}$ and $N_{pv}$ are the number of load and generation nodes respectively.

At the solution of the loadflow problem $\Delta P_i$ , $\Delta Q_i$ and $\Delta V_i$ for each node must be less than some preset tolerance, usually $10^{-3}$ in per-unit notation(pu). The loadflow problem may be formulated as a minimisation problem with the objective being to minimise the total squared mismatch which will eventually cause satisfaction of the $10^{-3}$ tolerance at each node. Such a minimisation problem is well suited to evolutionary computation methods such as Evolutionary Programming.

## 3 EVOLUTIONARY PROGRAMMING

Evolutionary programming is a stochastic optimisation technique based on the mechanics of evolution. It differs from classical calculus based methods in that it works with a population of solutions rather than a single solution. Within each iteration, a second population is produced through the *mutation* operator. This operator produces new solutions by perturbing the independent variables of an existing solution using a Gaussian random variable of mean zero and standard deviation $\beta$. The factor $\beta$ is referred to as the mutation factor and its value is critical in the EP process. Each individual is given a *fitness* score which is a measure of its optimality with respect to the objective function. A higher fitness is awarded to a more optimal solution.

Through the use of a *selection* scheme, these two populations are reduced to a single population containing the survivors of the competition. For the EP process to optimise, the candidate solutions with a higher fitness value must have a greater chance of survival in the selection scheme. This new population now undergoes the same process of mutation and competition until a stopping criterion is met.

## 4 EP LOADFLOW ALGORITHM

Based on the EP methodology, an algorithm for solving the loadflow can be established. The flow chart of the algorithm is shown graphically in Fig 1, and its main components are presented below.

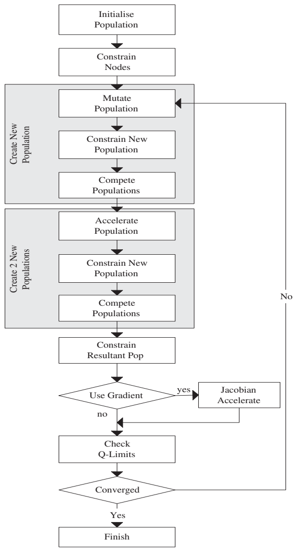


Figure 1: EP Loadflow flowchart

(i) *Individuals and Initialisation:* An individual in a population represents a candidate loadflow solution. The elements within the individual are the voltage magnitudes and phases for all nodes within the power system. Each of these elements in all candidates is independently initialised by a uniform random number.

(ii) *Fitness of Candidates:* A fitness score is assigned to each of the candidate solutions according to the relationship in (4).Using this method to assign fitness will lead to more optimal solutions having a higher fitness score.

$$f_i = \frac{10}{H_i + 0.0001} \tag{4}$$

where $f$ is the fitness of candidate $i$ and $H_i$ is the total squared mismatch of candidate $i$.

(iii) *Mutation:* At the mutation stage, the nodal voltages of each individual within the parent population undergo mutation in order to produce a new population. Typically the mutation standard deviation is formed by some relationship of the present individuals fitness, however, if it is possible to also include information which is specific to the problem being solved, the performance of the EP can be enhanced significantly. This approach is adopted in the present work where the mutation scheme of a nodal voltage depends on the type of the node, Load or Generation. A mutation scheme is now developed.

**Generation Nodes**: In the case of a generation node $i$, the only independent variable is that of the phase $\delta_i$. The phase is mutated as follows:

$$\delta_i' = \delta_i + N(0, \sigma_{pv}) \quad (5)$$

$$\sigma_{pv} = K_\delta \frac{f_{max} - f_i}{f_{max}} \quad (6)$$

where $\delta_i'$ is the phase in the new individual, $N(0, \sigma_{pv})$ is a Gaussian random number of mean 0 and standard deviation $\sigma_{pv}$. $K_\delta$ is a scaling constant for voltage phase. $f_{max}$ is the maximum fitness in the population at that iteration and $f_i$ is the fitness of the candidate being mutated.

**Load Nodes**: For a load node $i$, the independent variables are the voltage magnitude $|V_i|$ and the phase $\delta_i$. From the simple relationship between power flow in a transmission branch between nodes $i$ and $j$, we have the following equations:

$$P = \frac{V_i V_j}{X} \sin(\Delta\delta) \quad (7)$$

$$Q = \frac{V_j}{X} \left(V_i \cos(\Delta\delta) - V_j\right) \quad (8)$$

where $P, Q$ are the active and reactive powers flowing out of the receiving end ($j$) of the line. $\Delta\delta$ is the phase difference between the voltages at the end of the transmission line. $V_i$, $V_j$ are the voltages at nodes $i$ and $j$ respectively and $X$ is the reactance of the transmission line.

From these equations, it can be seen that the active power flow P, is strongly dependent on $\Delta\delta$ while the reactive power flow Q, is dependent on $|V_i| - |V_j|$. The nodal power is a summation of all the branch power flows as well as any local loads or devices and as such can be assumed to approximately follow these relationships also. The voltages at load nodes are hence mutated according to the following:

$$\delta_i' = \delta_i + N(0, \sigma_{\delta pq}) \quad (9)$$

$$V_i' = V_i + N(0, \sigma_{Vpq}) \quad (10)$$

$$\sigma_{\delta pq} = K_\delta \left(\frac{f_{max} - f_i}{f_{max}}\right) \left(\frac{\Delta P_i^2}{\Delta P_i^2 + \Delta Q_i^2 + 1}\right)$$

$$\sigma_{Vpq} = K_v \left(\frac{f_{max} - f_i}{f_{max}}\right) \left(\frac{\Delta Q_i^2}{\Delta P_i^2 + \Delta Q_i^2 + 1}\right)$$

where $K_v$ is a scaling constant for voltage magnitudes. In this way if the active power mismatch at a particular node is higher than the reactive power mismatch, the phase angle will be mutated more than the voltage magnitude. The opposite is true, if the reactive power mismatch is greater than the active power mismatch.

(iv) *Selection:* In order to form the resultant generation, the $k$ parent solutions $p_s$ , $s = 1, \ldots, k$ along with their corresponding offspring formed by mutation $p_s'$, $s = 1, \ldots, k$ each undergo a series of $N_t$ tournaments with randomly selected opponents. Each individual $s$, is assigned a score $\omega_{ps}$ according to:

$$\omega_{ps} = \sum_{i=1}^{N_t} \omega_t \quad (11)$$

$$\omega_t = \begin{cases} 1 & if f_{ps} > f_{pr} \\ 0 & otherwise \end{cases}$$

where $f_{ps}$ is the fitness of the individual under consideration and $f_{pr}$ is the fitness of the opponent $p_r$ . The opponent is chosen at random from the $2k$ individuals based on $r = \lfloor 2ku + 1 \rfloor$. $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, $u$ is a uniform random number in the interval [0,1]. The $k$ highest scoring candidate solutions are taken as the population in the next generation.

(v) *Termination:* The optimisation process is terminated when all of the nodal mismatches are within a tolerance of $10^{-3}$ pu.

## 5 CONSTRAINT SATISFACTION

The EP formulation given in Section 4 will not guarantee the satisfaction of the equality constraints for load-flow given in Equations (1) & (2). So although the EP will minimise the total squared mismatch, it may not produce a solution with all nodal mismatches within the preset tolerance of $10^{-3}$ pu. In order to aid the EP in finding the solution with near zero total squared mismatch, constraint satisfaction techniques must be used. The technique used to constrain the EP is presented in [8], where it was successfully applied to a genetic algorithm based loadflow method.

In order to satisfy the constraints described by equations (1) and (2), the voltages in the candidate solution are updated such as to produce a new candidate solution whose total squared mismatch is less than before and the nodes are closer to satisfying the constraints in (1) and (2). For each node $i$, the magnitude and phase parts of the voltage at node $d$ are calculated such that with this value for voltage at node $d$, the mismatch at node $i$ becomes zero. The derivation of these equations is lengthy and is omitted for clarity. They can be found in [8].

For each node $i$, there is a choice of changing one the remaining $N-1$ nodal voltages in order to provide a zero mismatch at node $i$. From this choice, the one which provides the lowest total squared mismatch is selected. The process is completed for all nodes in the system. Experimentation has shown that the resulting solutions are far superior when this technique is applied.

# 6 ACCELERATION TECHNIQUES

In order to improve the speed and reliability of the EP load-flow algorithm number of enhancements were made to the basic algorithm.

## 6.1 VOLTAGE UPDATE ORDER

In carrying out the constraint satisfaction as detailed in section 5, there are a number of choices of which nodes to update first. By experimentation, it was found that updating the nodes in order of greatest mismatch to lowest mismatch produced a faster rate of convergence. Also the number of voltages to be changed could be dramatically reduced. Consider a case where there are $N$ nodes, in applying the method of section 5, we are required to check a total of $(N-1)^2$ cases, this is computationally expensive and results in long calculation times.

It has been found through experimentation that it is sufficient to consider modifying only the voltage of the node presently being constrained. This reduces the number of cases to $N-1$. Experimentation has shown this method has close effectiveness to that detailed in section 5 but with a greatly reduced calculation time.

## 6.2 POPULATION ACCELERATION TECHNIQUE

A second technique [9] that was applied involves an intermediate re-mapping of the candidate solutions toward the currently best individual. The assumption behind this method is that the best individual in the population is closer to the global optimum than the others. By moving the population of solutions toward the best individual, it is hoped that by searching this region better solutions will be found.

Mathematically this is achieved by producing two additional populations (A & B) by moving the voltages of a candidate $i$ toward the voltages in the best individual as follows,

$$V'_{Aik} = 2V_k^{best} - V_{ik} \tag{12}$$

$$V'_{Bik} = V_{ik} + h(V_k^{best} - V_{ik}) \tag{13}$$

where $V'_{Aik}$ is the voltage of node $k$ of individual $i$ in the new population A, similarly for $V'_{Bik}$. $V_k^{best}$ is the value of the voltage at node $k$ in the best individual of the current population. $V_{ik}$ is the voltage at node $k$ in the $i^{th}$ individual in the current population. $h$ is a uniform random number between -1 and 1. These three populations are then resolved into a resultant population using the selection scheme as with mutation.

## 6.3 GRADIENT ACCELERATION

The basic EP algorithm does not require any gradient information, however when the dimensions of the problem become large as in the case of loadflow, evolutionary computation techniques may require excessive iterations or large population sizes in order to find a solution where possible. To overcome this the gradient may be used to accelerate individuals within the population towards the global optimum.

This is achieved with the use of the Jacobian matrix. At the step *Jacobian Acceleration* of Fig.1, a pre-set percentage of candidates in the population undergo a gradient step as used in the Newton-Raphson method. In this way, the EP acts as a global optimisation framework with the Jacobian providing a local search mechanism. If the Jacobian becomes singular(in which case the Newton-Raphson method fails) the gradient step is simply omitted until it is no longer singular.

# 7 APPLICATION RESULTS

The developed algorithm has been applied to a number of standard loadflow problems with great success. Excluding the use of the gradient information, the method has found solutions for systems as large as 57 nodes. When the gradient information is included, the node limitation is removed and the EP shows significant enhancements over the basic Newton-Raphson method.

To demonstrate the features of the algorithm, results for the IEEE 30 node test system are presented below along with the IEEE 57 node systems when gradient enhancement is used. All trials were run on an Intel Pentium II 450 computer using the C programming language.

### 7.1 IEEE 30 NODE SYSTEM

The standard IEEE 30 node test system was solved using the developed algorithm. The system consists of 30 nodes with 6 being generation nodes with reactive power limits, providing a problem with 53 variables and 63 inequality constraints to be satisfied at the solution. In all cases the population size was set at 50 while the maximum iterations executed is fixed at 50. $K_V$ and $K_\delta$ were set at 0.2 and $0.35^{rad}$ respectively in all cases. The problem was solved 100 times and produced the correct result in 92 of the 100 trials. The average number of iterations required was 19, while the time taken per iteration was 0.344 seconds.

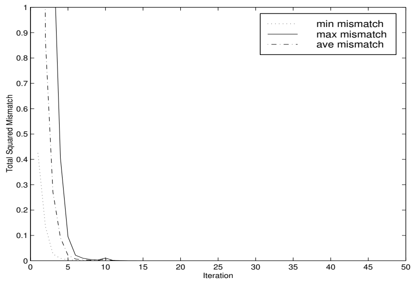


Figure 2: Convergence on the IEEE 30 system

To demonstrate the convergence of the algorithm, the minimum, maximum and average total mismatch at each iteration, averaged over the 100 trials is plotted below in Fig.2. From this, it can be seen that the total squared mismatch is rapidly minimised initially in the first 10 iterations. Following this, convergence slows as the nodes are close to their solution point and the relative change in mismatch is small while the remaining constraints are being satisfied.

#### 7.1.1 POPULATION ACCELERATION

The population acceleration technique described in section 6 has a dramatic effect on the performance of the loadflow algorithm. When the 30 node test system was run 100 times without the inclusion of the acceleration technique, the algorithm failed to converge to a solution in all trials. As can be seen by comparing the convergence without acceleration shown in Fig.3, without population acceleration the convergence is much slower. The large spike at the $10^{th}$ iteration is a result of initiating reactive power limit switching at that iteration. Without acceleration, the disturbance created by switching is much larger as the solution is far from being converged.

#### 7.1.2 EFFECTIVENESS OF MUTATION

To demonstrate the effectiveness of the mutation scheme developed in section 4, the algorithm was run 100 times with a basic EP mutation scheme given below in (14). The average minimum total squared mismatch at each iteration for the case with the developed mutation scheme and without it are plotted below in Fig.4.

When the developed mutation scheme is included the convergence is much faster as can be seen in Fig.4. It is also much more consistent in finding the solution with a correct result found in 92% of the trials with the developed mutation scheme and only 78% in the case where simple mutation is used. This result demonstrates the effectiveness of including problem specific information in the EP mutation.

$$\begin{aligned} \sigma_{\delta pq} &= K_\delta \left( \frac{f_{max} - f_i}{f_{max}} \right) \\ \sigma_{Vpq} &= K_V \left( \frac{f_{max} - f_i}{f_{max}} \right) \end{aligned} \tag{14}$$

### 7.2 IEEE 57 NODE SYSTEM

The algorithm was next applied to the IEEE 57 node test system which comprises of 7 generation nodes each with reactive power limits. There are 106 variables to be determined and 118 inequality constraints to be satisfied at the solution. The developed algorithm was run 100 times to solve this system and found a solution

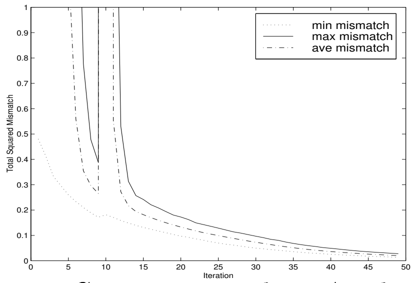


Figure 3: Convergence without Acceleration

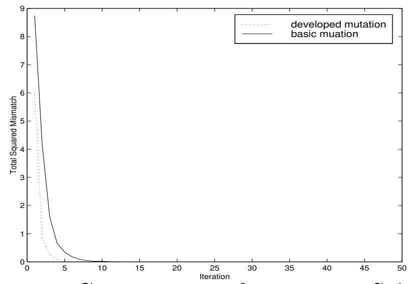


Figure 4: Comparison of Mutation Schemes

in 50 cases, this performance is insufficient for loadflow applications. As the dimension of the problem become larger, evolutionary programming EP techniques can have trouble in finding the solution, especially when the number of constraints is also high. The average convergence of the algorithm is plotted below in Fig.5, the effect of system size on convergence can be seen by comparing this case with that of the 30 node case in Fig.2. The algorithm parameters in both cases were identical. In order to overcome this limitation it is

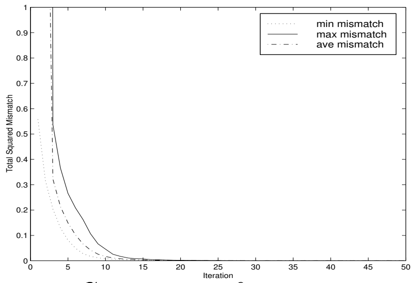

Figure 5: Convergence for IEEE 57 system

necessary to incorporate more information about the loadflow problem into the EP structure. As the gradient information in the loadflow is well researched and developed and intermediate gradient step was implemented in the EP as described in section 6. Although this breaks with the non-gradient nature of Evolutionary Programming, the resulting composite algorithm is both powerful and robust. When run on the IEEE 57 node system convergence occurred in 2 iterations and a solution was found in 100% of the trials, the percentage of the population undergoing gradient acceleration was set at 50%.

### Conclusions

An evolutionary programming based loadflow algorithm has been developed. The proposed method has been enhanced with a mutation scheme which contains loadflow specific information, the result of this have been validated on standard test systems. Population acceleration has also been incorporated into the technique with large performance gains in both reliability and convergence speed. Finally the algorithm includes gradient information about the loadflow in order to solve larger and more complex systems.

The paper demonstrates that evolutionary programming is a powerful optimisation tool whose performance can be greatly enhanced with the inclusion of problem specific information. The EP method remains the global optimisation technique which controls the progression of the population using acceleration, constraint satisfaction and gradient information. The work presented in this paper is ongoing and further improvements to the method are expected.

### Acknowledgements

The authors wish to express their sincere gratitude to the Western Power Corporation, Western Australia, and to the Australian Research Council for their support to the work reported in the paper. They are also grateful to Phillip Kelloway of Western Power Corporation for his support given to the present project. Yuryevich acknowledges the award of an Australian Postgraduate Award (Industry) from the Department of Employment Education and Training and the Australian Research Council.

## References

[1] K.P. Wong and J. Yuryevich. Evolutionary programming-based economic dispatch for environmentally constrained economic dispatch. *IEEE Trans. on Power Systems*, 13(2):301–306, 1997.

[2] S.A. Kazarlis, A.G. Bakirtzis, and V. Petrdis. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. on Power Systems*, 11(1):372–380, 1995.

[3] J. Yuryevich and K.P. Wong. Evolutionary programming based optimal power flow algorithm. Paper PE-137-PWRS-0-10-1998, to appear in IEEE Trans. on Power Systems 1999.

[4] J.J. Grainger and W.D. Stevenson. *Power System Analysis.* McGraw-Hill, 1994.

[5] B. Stott and O. Alsac. Fast decoupled load flow. *IEEE Trans. on PAS*, 93:859–869, 1974.

[6] D.B. Fogel. *Evolutionary Computation: Toward a new Philosophy in Machine Intelligence.* IEEE Press, 1995.

[7] T. Back. *Evolutionary algorithms in theory and practice.* Oxford University Press, 1996.

[8] K.P. Wong, A. Li, and M.Y. Law. Development of constrained genetic algorithm load flow method. *IEE Proc. Gener. Transm. and Distrib.*, 144(2):91–99, 1997.

[9] K.P. Wong and A. Li. A technique for improving the convergence characteristic of genetic algorithms and its application to a genetic-based load flow algorithm. In *Lecture Notes in Artificial Intelligence 1285*, pages 167–176. Springer-Verlag, 1996.