

# Survey of Reactive Power Planning Methods

Wenjuan Zhang, *Student Member, IEEE*, Leon M. Tolbert, *Senior Member, IEEE*

**Abstract** — Reactive power planning (RPP) involves optimal allocation and determination of the types and sizes of the installed capacitors. Traditionally, the locations for placing new VAr sources were either simply estimated or directly assumed. Essentially, it is a large-scale nonlinear optimization problem with a large number of variables and uncertain parameters. There are no known ways to solve such Nonlinear Programming Problems (NLP) exactly in a reasonable time. This paper introduces mathematical formulations and the advantages and disadvantages of nine categories of methods, which are divided into two groups, conventional and advanced optimization methods. The first group of methods is often trapped by a local optimal solution; the second one can *guarantee* the global optimum but needs more computing time. The corresponding development history period is also split into a conventional period and an artificial intelligence period. In the second period, there is a tendency to integrate several approaches to solve the RPP problem.

**Index Terms** — reactive power planning, reactive power optimization, optimal power flow

## I. INTRODUCTION

**M**ORE than 25 years ago, Carpentier introduced a generalized, nonlinear mathematical programming formulation of the economic dispatch problem including voltage and other operating constraints, which was later named the optimal power flow (OPF) problem. Today any problem that involves the determination of the instantaneous “optimal” steady state of an electric power system is an optimal power flow problem. Reactive power planning (RPP) is a typical OPF problem.

The OPF problem has the following characteristics:

1. An OPF not only addresses transient and dynamic stability, but also steady-state operation of the power system.
2. OPF is not a mathematically convex problem, so the valuable property of the convex function, that any local optimal is also globally optimal, can not be used; as a result, most general nonlinear programming techniques might converge to a local minimum instead of a unique global minimum.
3. A power system OPF analysis can have many different goals and corresponding objective functions; usually, the problem is to minimize the fuel cost and/or system losses, taking into account the power flow constraints imposed by the transmission network together with

other constraints such as real and reactive power generation limits and voltage magnitude limits.

The following is a classical formulation of an OPF problem, whose goal is to minimize the total cost of real and reactive generation.

$$\min_{P_g, Q_g, Q_{ci}} \sum_{i \in N_g} [f_{1i}(P_{gi}) + f_{2i}(Q_{gi})] + \sum_{i \in N_c} [C_{fi} + C_{ci} \times Q_{ci}] \times r_i$$

subject to

$$P_{gi} - P_{Li} - P(V, \theta) = 0 \text{ (active power balance equations)}$$

$$Q_{gi} - Q_{Li} - Q(V, \theta) = 0 \text{ (reactive power balance equations)}$$

$$S_{ij}^f \leq S_{ij}^{\max} \text{ (apparent power flow limit of lines, from side)}$$

$$S_{ij}^t \leq S_{ij}^{\max} \text{ (apparent power flow limit of lines, to side)}$$

$$V_i^{\min} \leq V_i \leq V_i^{\max} \text{ (bus voltage limits)}$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max} \text{ (active power generation limits)}$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max} \text{ (reactive power generation limits)}$$

$$Q_{ci}^{\min} \leq Q_{ci} \leq Q_{ci}^{\max} \text{ (VAr source installation limits)}$$

where  $f_{1i}$  and  $f_{2i}$  are the costs of active and reactive power generation, respectively, for generator  $i$  at a given dispatch point.  $C_{fi}$  is the fixed VAr source installation cost, and  $C_{ci}$  is the per unit VAr source purchase cost.  $Q_{ci}$  is VAr source installed at bus  $i$ .  $N_g$  is the set of generator buses;  $N_c$  is the set of possible VAr source installment buses.  $r_i$  equals 1 if there is installation of reactive power source at bus  $i$ , otherwise, it is zero.

The solution techniques of OPF have evolved over many years, and dozens of approaches have been developed, each with its particular mathematical and computational characteristics. However, real-life OPF problems are much more complicated than their classical formulations. OPF methods vary considerably in their adaptability to the modeling and solution requirements of different engineering applications. Therefore, there has been no single formulation and solution approach that suits a wide range of OPF-based calculations. The majority of the techniques discussed in the literature of the last 20 years use at least one of the following 9 categories of methods, which also include some subcategories.

- Nonlinear Programming (NLP)
- Linear Programming (LP)
- Mixed Integer Programming (MIP)
- Decomposition Method
- Heuristic Method (HM)
- Simulated Annealing (SA)
- Evolutionary Algorithms (EAs)
- Artificial Neural Network (ANN)
- Sensitivities Analysis

W. Zhang and L. M. Tolbert are with the Department of Electrical and Computer Engineering, The University of Tennessee, Knoxville, TN 37996-2100 USA (e-mail: wzhang5@utk.edu, tolbert@utk.edu). This work was supported in part by the National Science Foundation under Contract NSF ECS-0093884.

This work divides all of the above methods into two groups depending on whether they can find the global optimal solution; one group has conventional optimization methods including Nonlinear Programming, Linear Programming, Mixed Integer Programming, Decomposition Method, and Heuristic Method; the other involves artificial intelligence methods, also named advanced optimization methods including Simulated Annealing, Evolutionary Algorithms, and Artificial Neural Networks.

Many of the existing conventional optimization algorithms for the VAR sources planning problem employ various greedy search techniques — accepting only changes that yield immediate improvement, which are effective for the optimization problems with deterministic quadratic objective function that has only one minimum. However, it often induces local optima rather than global optima, and sometimes results in divergence when minimizing both objective functions at the same time. As a result, these solution algorithms usually achieve local minima rather than global minima.

Recently, new advanced optimal methods based on artificial intelligence such as SA, EAs, and ANNs have been used in RPP to deal with local minimum problems and uncertainties. Increasingly, these methods are being combined with conventional and advanced optimization methods to solve the RPP problem. Corresponding to the methods division of two groups, the development of RPP problem can be divided into two periods, a “conventional period” and an “advanced period”; the following will introduce the characteristics of each period and method.

## II. CONVENTIONAL PERIOD METHODS

### A. Nonlinear Programming (NLP)

#### 1) Generalized Reduced Gradient (GRG)

To solve a nonlinear programming problem, the first step in this method is to choose a search direction in the iterative procedure, which is determined by the first partial derivatives of the equations (the reduced gradient), so such methods will be referred to here as first-order methods. These OPF implementations characteristically suffered from two major problems when applied to large-sized power systems: (1) Even though it has global convergence, which means the convergence can be guaranteed independent of the starting point, a slow convergent rate occurred because of zigzagging in the search direction, which means the solution trajectory hits one of the constraint boundaries in successive iterations without achieving a substantial decrease in the value of the objective function. (2) Different “optimal” solutions were obtained depending on the starting point of the solution because the method can only find a local optimal solution.

Extending the reduced gradient method to handle nonlinear constraints is referred to as the generalized reduced gradient method, and it is briefly sketched below [1].

Consider the following problem:

$$\text{Minimize } f(x)$$

$$\text{Subject to } Ax = b \quad x \geq 0$$

Note that  $A$  can be decomposed into  $[B, M]$  corresponding to the decomposition of  $x'$  into  $[x'_B, x'_N]$ . Here  $x_B$  is called the basic vector,  $x_N$  are the components of the nonbasic vector, and  $B$  is an  $m \times m$  invertible matrix.  $\nabla f(x)' = [\nabla_B f(x)', \nabla_N f(x)']$ , where  $\nabla_B f(x)$  is the gradient of  $f$  with respect to the basic vector  $x_B$ . Recall that a direction  $d$  is a descent direction of  $f$  at  $x$ , if  $\nabla f(x)' d < 0$ .

Main steps:

1. A direction vector  $d'_k = [d'_B, d'_N]$  was specified, where  $d_N$  and  $d_B$  are obtained from the following:

$$I_k = \text{index set of the } m \text{ largest components of } x_k$$

$$r' = \nabla f(x_k)' - \nabla_B f(x_k)' B^{-1} A$$

$$d_j = -r_j \text{ if } j \notin I_k \text{ and } r_j \leq 0$$

$$\text{or } d_j = -x_j r_j \text{ if } j \in I_k \text{ and } r_j > 0$$

$$d_B = -B^{-1} N d_N$$

2. A line search is performed along  $d_k$

$$\text{Minimize } f(x_k + \lambda d_k)$$

$$\text{Subject to } 0 \leq \lambda \leq \lambda_{\max}$$

$$\text{Let } x' = x_k + \lambda_k d_k$$

3. In the original problem, suppose  $Ax = h(x)$ ; since  $h(x) = 0$  is not necessarily satisfied, we need a correction step. Toward this end, the Newton-Raphson method is then used to obtain  $x_{k+1}$  satisfying  $h(x_{k+1}) = 0$ . Replace  $k$  by  $k+1$  and repeat step 1.

Wu et al. [2] developed an algorithm to incorporate certain features of the generalized reduced gradient and the penalty function methods. At each iteration, a modified reduced gradient is used to provide a direction of the movement.

#### 2) Newton's Approach

The successive quadratic programming and Newton's method require the computation of the second partial derivatives of the power flow equations and other constraints (the Hessian) and are therefore called second-order methods. For an objective function of  $f(x)$ , Newton's method defines the search direction  $d_k = -H(x_k)^{-1} \nabla f(x_k)$ , where  $H(x_k)$  is the Hessian matrix of  $f(x)$  at  $x_k$ . The successor point is  $x_{k+1} = x_k + \lambda d_k$ , where  $\lambda$  is the step length. If we can properly choose  $\lambda$ , this method can guarantee convergence regardless of the starting point, which means a global convergence. Some quasi-Newton methods, e.g. BFGS (Broyden-Fletcher-Goldfarb-Shanno) have global convergence. Newton's method is developed for the Optimal Power Flow in [3, 4].

### 3) Successive Quadratic Programming (SQP)

SQP methods also known as sequential, or recursive, quadratic programming, employ Newton's method (or quasi-Newton methods) to directly solve the Karush-Kuhn-Tucker (KKT) conditions for the original problem. As a result, the accompanying subproblem turns out to be the minimization of a quadratic approximation to the Lagrangian function optimized over a linear approximation to the constraints. To present the concept of this method, consider the following nonlinear problem (all functions are assumed to be continuously twice-differentiable):

$$\begin{aligned} \text{P: Minimize } & f(x) \\ \text{Subject to } & h_i(x) = 0 \quad i = 1, \dots, l \end{aligned}$$

Defining  $\nabla^2 L(x_k) = \nabla^2 f(x_k) + \sum_{i=1}^l v_{ki} \nabla^2 h_i(x_k)$  to be the usual Hessian of the Lagrangian at  $x_k$  with the Lagrange multiplier vector  $v_k$ , and letting  $\nabla h$  denote the Jacobian of  $h$ . The quadratic minimization subproblem is stated below [1].

$$\begin{aligned} \text{QP: Minimize } & f(x_k) + \nabla f(x_k)' d + \frac{1}{2} d' \nabla^2 L(x_k) d \\ \text{Subject to } & h_i(x_k) + \nabla h_i(x_k)' d = 0, \quad i = 1, \dots, l \end{aligned}$$

The main step is as follows. First, solve the quadratic subproblem QP to obtain a solution  $d_k$  along with a set of Lagrange multipliers  $v_{k+1}$ . Second, check whether  $(x_k, v_{k+1})$  satisfies the KKT conditions. Otherwise, set  $x_{k+1} = x_k + d_k$ , and repeat the main step.

The objective function of QP represents not just a quadratic approximation for  $f(x)$  but also incorporates an additional term  $\frac{1}{2} \sum_{i=1}^l v_{ki} d' \nabla^2 h_i(x_k) d$  to represent the curvature of the constraints. In fact, defining the Lagrangian function  $L(x) = f(x) + \sum_{i=1}^l v_{ki} h_i(x)$ , the objective function of QP can alternatively be written as

$$\text{Minimize } L(x_k) + \nabla L(x_k)' d + \frac{1}{2} d' \nabla^2 L(x_k) d$$

This supports the quadratic convergence rate behavior in the presence of nonlinear constraints. The quadratic convergence rate is faster than the superlinear convergence rate, and the superlinear convergence rate is faster than the linear convergence rate. Therefore, the SQP method is better than the GRG method as far as convergence rate is concerned. In addition, a properly implemented second-order OPF solution method is robust with respect to different starting points, which means it has a global convergence characteristic.

Papalexopoulos et al. [5] and Kermanshahi [6] proposed to solve the reactive power OPF problem by the successive quadratic programming (SQP) method. Rau [7] proposed a radial equivalent mapping method to map the complex meshed electrical network to radial network equivalents to mirror the transactions. Such equivalent networks do not replace the existing power flow analyses, simultaneous feasibility tests, or other procedures. However, the equivalents obtained by using quadratic programs retain the

properties and laws of physics of the original network while offering a convenient method of tracking market transactions to allocate costs and responsibilities. Consequently, the minimization problem is a classical Quadratic Program (QP).

### B. Linear Programming (LP)

Consider the following primal linear program:

$$\begin{aligned} \text{Minimize } & cx \\ \text{Subject to } & Ax = b \quad x \geq 0 \end{aligned}$$

The dual problem can be stated as follows:

$$\begin{aligned} \text{Maximize } & wb \\ \text{Subject to } & wA \leq c \end{aligned}$$

Thus, in the case of linear programs, the dual problem does not involve the primal variables, furthermore, the dual problem itself is a linear program.

Mamandur and Chenoweth [8] developed an efficient algorithm based on loss minimization while satisfying the network performance constraints on control variables. The method uses a dual LP technique to determine the optimal control variables' adjustment, simultaneously satisfying the constraints.

Venkataramana et al. [9] applied LP in combination with parametric programming to find the minimum amount of required reactance. Papalexopoulos et al. [5] presented a fast and reliable method for linear reactive power optimization using a decoupling technique to reduce the problem into several sub-problems, in which the active power OPF problem was solved by a successive LP method.

Iba et al. [10] presented an LP-based algorithm suitable for both loss minimization and investment cost minimization. The linearized objective function consists of two parts. One represents the effect of reductions in power losses, and the other represents investment costs of new capacitor or reactor banks.

Yehia et al. [11] proposed an LP trade-off methodology as a decision support tool for an optimal distribution based on the full integration of both technical and economic aspects of the problem. A modified simplex method is adopted to solve the LP problem.

Thomas et al. [12] proposed another LP-based technique, which is to linearize the nonlinear reactive optimization problem with respect to the voltage variables. With consideration of the effect of contingencies, the LP-based technique has led to a general purpose security-constrained optimal power flow solver (SC-OPF), which can handle realistic problems satisfactorily.

Venkatesh et al. [13] developed the successive multiobjective fuzzy LP (MFLP) framework to solve the nonlinear programming problem of ORPP. In the MFLP, each of the objectives and constraints is expressed as a fuzzy set and is assigned a satisfaction parameter.

The LP approach has several advantages. One is the reliability of the optimization, especially the convergence properties. Another is its ability to recognize problem infeasibility quickly. Third, the approach accommodates a

large variety of power system operating limits, including the very important contingency constraints. The nonseparable loss-minimization problem can now be solved, giving the same results as NLP on power systems of any size and type. Former limitations on the modeling of generator cost curves have been eliminated.

Nevertheless, while the LP approach has a number of important attributes, its range of application in the OPF field has remained somewhat restricted. Even for recent studies, there are some drawbacks such as: inaccurate evaluation of system losses, high possibility to be trapped in the local optimal, and insufficient ability to find an exact solution.

### C. Mixed Integer Programming (MIP)

The following is the formulation of a mixed integer program:

$$\begin{aligned} \text{Minimize} \quad & z = c_1x + c_2y \\ \text{Subject to} \quad & A_1x + A_2y \geq b \\ & x, y \geq 0 \quad y \text{ is an integer} \end{aligned}$$

Theoretically the VAR planning problem can be formulated as a nonlinear mixed-integer programming optimization method with 0-1 integer variables representing whether new VAR control devices should be installed.

Aoki et al. [14] presented an optimization approach, which is based on a recursive mixed-integer programming technique using an approximation method. Since there are no general mathematical solutions to nonlinear mixed-integer programming problems, approximations have to be made. In the nonlinear mixed-integer programming problem formulation, both the number and value of capacitors were still treated as continuously differentiable. A decomposition technique was then employed to decompose the problem into a continuous problem and an integer problem. However, such a procedure makes the algorithms rather complex.

### D. Decomposition Method

Decomposition methods can greatly improve the efficiency in solving a large-scale network by reducing the dimensions of the individual subproblems; the results show a significant reduction of the number of iterations, required computation time, and the memory space. Also, decomposition permits the application of a separate method for the solution of each subproblem, which makes the approach quite attractive.

#### 1) Dantzig-Wolfe Decomposition Method (DWD) or Dual-Decomposition for LP

The DWD was developed to decompose a system's optimization problem into several subproblems corresponding to specific areas in the power system, so this method is very fast. The solution of the master problem depends on the dual solutions provided by the subproblems. The master problem passes down a new set of cost coefficients to the subproblem and receives a new column based on these cost coefficients.

The DWD algorithm is as follows [15]. Consider the linear program P, where  $X$  is a polyhedral set representing constraints of a special structure. For convenience, let us assume that  $X$  is nonempty and bounded in the following decomposition algorithm, then any point  $x \in X$  can be represented as a convex combination of the finite number of extreme points  $x_j$  of  $X$  as

$$\begin{aligned} x &= \sum_{j=1}^t \lambda_j x_j \\ \sum_{j=1}^t \lambda_j &= 1 \quad \lambda_j \geq 0 \quad j = 1, \dots, t \end{aligned}$$

Then, P is transformed into the Master problem:

$$\begin{aligned} \text{P: Minimize} \quad & cx \\ \text{Subject to} \quad & Ax = b \quad x \in X \end{aligned}$$

Master Problem (MP):

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^t (cx_j) \lambda_j \\ \text{Subject to} \quad & \sum_{j=1}^t (Ax_j) \lambda_j = b \\ & \sum_{j=1}^t \lambda_j = 1 \quad \lambda_j \geq 0 \quad j = 1, \dots, t \end{aligned}$$

Given a basic feasible solution  $(\lambda_B, \lambda_N)$  with dual variables corresponding to the above two equality constraints by  $w$  and  $\alpha$ , the following subproblem is easier to solve than the master problem.

Subproblem:

$$\begin{aligned} \text{Maximize} \quad & (wA - c)x + \alpha \\ \text{Subject to} \quad & x \in X \end{aligned}$$

Deeb and Shahidehpour [16] applied this algorithm to a large-scale power network. The proposed method is robust, not sensitive to the feasibility of the starting point, nor to the type and size of variables in different areas.

#### 2) Lagrange Relaxation Decomposition Method (LRDM)

The previous dual-decomposition algorithm solves a dual subproblem and a master problem until the optimum solution is achieved. Geoffrion [17] showed that this method can be generalized via Lagrangean relaxation to deal with Mixed Integer Programming (MIP), which is called the Lagrangean relaxation decomposition method (LRDM). The LRDM algorithm is as follows [15]:

$$\begin{aligned} \text{P: Minimize} \quad & cx \\ \text{Subject to} \quad & Ax = b \\ & x \in X = \{x : Dx \geq d, x \geq 0\} \end{aligned}$$

We designate  $w$  as the dual variables associated with the constraints  $Ax = b$ . The following master problem (MP) is known as a Lagrangian dual to problem P.

$$\begin{aligned} \text{MP: Maximize} \quad & \{\theta(w) : w - \text{unrestricted}\} \\ \text{Subproblem:} \quad & \theta(w) = wb + \text{Min}\{(c - wA)x : x \in X\} \end{aligned}$$

### 3) Benders Decomposition Method (BDM)

The classical Benders method can be applied to linear disjunctive programming problems, and the generalized Benders method to nonlinear disjunctive problems. The beauty of the BDM is that one can often distinguish a few “hard” variables  $y$  from many “easy” variables  $x$ , and when the hard variables are fixed, the problem becomes easy by decoupling it into several small problems that can be solved separately. The BDM algorithm is as follows [15]:

$$\begin{aligned} \text{P: Minimize} \quad & cx \\ \text{Subject to} \quad & Ax = b \\ & x \in X = \{x : Dx \geq d, x \geq 0\} \end{aligned}$$

We designate  $w$  as the dual variables associated with the constraints  $Ax = b$ .

$$\begin{aligned} \text{MP: Maximize} \quad & z \\ \text{Subject to} \quad & z \leq wb + (c - wA)x_j \quad \text{for } j = 1, \dots, t \\ & z, w \text{ -unrestricted} \end{aligned}$$

$$\text{Subproblem: } \overline{wb} + \underset{x \in X}{\text{Minimum}}\{(c - wA)x\}$$

Where,  $(\overline{z}, \overline{w})$  is an optimal solution for a relaxed master problem. If  $\overline{z}$  is less than or equal to the optimal objective value of the subproblem, then we are done. Otherwise, assuming  $x_k$  solves the subproblem, we can generate the constraint  $z \leq wb + (c - wA)x_k$  and add it to the current relaxed master program and reoptimize.

Gomez et al. [18] extended the Benders approach to represent normal and contingency operating conditions of a power system. When a contingency happens, this methodology can provide flexible, variable control of corrective and preventive actions.

### 4) Generalized Benders Decomposition (GBD)

LP, NLP, and MIP all treat the number and the value of the capacitors as continuously differentiable. The favorite approach is to decompose the problem into two optimization problems: (1) the master subproblem dealing with the investment decision of installing discretized new VAR devices and (2) the slave problem dealing with the operation optimization. These techniques normally use the Generalized Bender’s Decomposition (GBD) Method to decompose it into a continuous problem and an integer problem. However, the GBD technique does not always perform well in solving practical VAR problems; the convergence can only be guaranteed under some convexity assumptions of the objective functions of the operation subproblem, so the solution cannot always be guaranteed.

Consider the generalization of BDM for the following nonlinear and/or discrete problem [15].

$$\begin{aligned} \text{P: Minimize} \quad & cx + f(y) \\ \text{Subject to} \quad & Ax + By = b \\ & x \geq 0 \quad y \in Y \end{aligned}$$

where  $c, b, x, y$  are vectors,  $A, B$  are matrices,  $f$  is an arbitrary function, and  $Y$  is an arbitrary set.

$$\begin{aligned} \text{MP: Minimize} \quad & z \\ \text{Subject to} \quad & z \geq f(y) + w_j(b - By) \quad \text{for } j = 1, \dots, t \\ & d_j(b - By) \leq 0 \quad \text{for } j = 1, \dots, l \\ & y \in Y \end{aligned}$$

where  $w_1, \dots, w_l$  and  $d_1, \dots, d_l$  are the extreme points and extreme directions generated so far.

$$\begin{aligned} \text{Subproblem: Maximize} \quad & w(b - By) \\ \text{Subject to} \quad & wA \leq c \end{aligned}$$

The set  $Y$  can be discrete, and so the procedure can be used for solving mixed integer problems.

Hong et al. [19] integrated the Newton-OPF with the GBD to solve the long term VAR planning problem. Abdul-Rahman [20] extended the GBD and combined it with the fuzzy set. Furthermore, the operation problem is decomposed into 4 subproblems via Dantzig-Wolfe Decomposition (DWD), leading to a significant reduction in its dimensions for personal computer applications by applying a second DWD to each subproblem. Yorino et al. [21] applied GBD to decouple a mixed integer nonlinear programming problem of a large dimension taking into account the expected cost for voltage collapse and corrective controls.

### 5) Cross Decomposition Algorithm (CDA)

Deeb [22] introduced a novel Cross Decomposition Algorithm (CDA), which is based on an iterative process between the subproblems of the BDM and the Lagrangean Relaxation Decomposition Method (LRDM). This method has shown its superiority through computational time and memory space reductions by limiting the optimization iterations between the primal and dual subproblems.

### E. Heuristic Method (HM)

Heuristic means a simplification or educated guess that reduces or limits the search for solutions in domains that are difficult and poorly understood. Unlike algorithms, heuristics do not guarantee optimal or even feasible solutions and are often used with no theoretical guarantee. The heuristic approach used in the reactive power planning can be defined as “bound reduction heuristics,” which is simple, yet robust.

Mantovani [23] proposed to use linear programming and binary search in conjunction with a heuristic process designed to convert an initially relaxed solution into a discrete one. A binary search is a searching algorithm, which works on a sorted table by testing the middle of an interval, eliminating the half of the table in which the key cannot lie, and then repeating the procedure iteratively. Each final trial solution of the binary tree can be represented by a “pattern” of the type (010010...010001) where 0 means that reactive sources can not be installed in the corresponding bus and 1 means that such installation is permitted.

### III. ADVANCED PERIOD METHODS

#### A. Simulated Annealing (SA)

From a mathematical standpoint, simulated annealing—as introduced by Kirkpatrick, Gelatt, and Vecchi (1983)—is a stochastic algorithm aimed at minimizing numerical functions of a large number of variables, and it allows random upward jumps at judicious rates to provide possible escapes from local energy wells. As a result, it converges asymptotically to the global optimal solution with probability one.

Simulated annealing is analogous to the annealing process used for crystallization in physical systems. In order to harden steel, one first heats it up to a high temperature not far away from the transition to its liquid phase. Subsequently one cools down the steel more or less rapidly.

This process is known as annealing. According to the cooling schedule the atoms or molecules have more or less time to find positions in an ordered pattern. The highest order, which corresponds to a global minimum of the free energy, can be achieved only when the cooling proceeds slowly enough. Otherwise, the frozen status will be characterized by one or the other local minima only. That is why the successful applications of simulated annealing require careful adjustments of temperature cooling schedules as well as proper selection of neighborhoods in the configuration space. A fact well-supported by recent mathematical results is that simulated annealing algorithms are quite greedy in computing time.

SA is a powerful general-purpose technique for solving combinatorial optimization problems, which can be described as [24]:

Given a finite configuration space  $S = \{x \mid x = (x_1, x_2, \dots, x_m)\}$ , and a cost function  $C: S \rightarrow R$ , which assigns a real number to each configuration, we want to find an optimum configuration  $x^* \in S$ , such that  $\forall y \in S, C(x^*) \leq C(y)$ .

The SA algorithm can be described as follows:

Given a general finite configuration space  $S$ , an energy function  $U$  on  $S$ , a cooling schedule  $(T_n)$ , and the exploration matrix  $q$ . Therefore, we fix a symmetric neighborhood system  $V(i), i \in S$ , where  $V(i)$  is the set of neighbors of configuration  $i$ . A generic sequential annealing algorithm is presented on  $S$ , that generates a random sequence  $X_n \in S$  of configurations that will tend to concentrate, as  $n \rightarrow \infty$ , on the set of absolute minima of  $U$ . The sequence  $X_n$ , with arbitrary initial configuration  $X_0$ , is then a Markov chain that is nonhomogeneous in time, with its transition function defined by

$$P(X_{n+1} = j \mid X_n = i) = p_{T_n}(i, j),$$

where

$$p_T(i, j) = q(i, j) \exp\left[-\frac{1}{T}(U_j - U_i)^+\right] \quad \text{if } j \neq i$$

$$p_T(i, i) = 1 - \sum_{j \neq i} p_T(i, j)$$

$$q(i, j) = \frac{1}{\text{card}(V_i)} \quad \text{for } j \in V_i$$

$$q(i, j) = 0 \quad \text{for } j \notin V_i$$

Hsiao et al. [25] provided a methodology based on simulated annealing to determine the location to install VAR sources, the types and sizes of VAR sources to be installed, and the settings of VAR sources at different loading conditions. In order to speed up the solution algorithm, a slight modification of the fast decoupled load flow is incorporated into the solution algorithm.

In [26], Hsiao et al. formulated the VAR planning problem as a constrained, multi-objective and non-differentiable optimization problem and provided a two-stage solution algorithm based on the extended simulated annealing technique and the  $\mathcal{E}$ -constraint method. However, only the new configuration (VAR installation) is checked with the load flow, and existing resources such as generators and regulating transformers are not fully explored. In addition, the SA takes much CPU time to find the global optimum.

Then Jwo et al. [27] proposed a hybrid expert system/SA method to improve the CPU time of SA while retaining the main characteristics of SA. Expert system simulated annealing (ESSA) is introduced to search for the global optimal solution considering both quality and speed at the same time. They use an expert system consisting of several heuristic rules to find a local optimal solution, which will be employed as an initial starting point of the second stage. This method is insensitive to the initial starting point, and so the quality of the solution is stable. It can deal with a mixture of continuous and discrete variables.

#### B. Evolutionary Algorithms (EAs)

Natural evolution is a population-based optimization process. An EA is different from conventional optimization methods; it does not need to differentiate cost function and constraints. Theoretically, EAs converge to the global optimum solution with probability one. EAs, including evolutionary programming, evolutionary strategy, and genetic algorithms, are artificial intelligence methods for optimization based on the mechanics of natural selection, such as mutation, recombination, reproduction, crossover, selection, etc.

Mutation randomly perturbs a candidate solution; recombination randomly mixes their parts to form a novel solution; reproduction replicates the most successful solutions found in a population; crossover exchanges the genetic information of two strings that are selected from the population at random; whereas selection purges poor solutions from a population. Each method emphasizes a different facet of natural evolution.

Evolutionary programming stresses behavioral change at the level of the species. Evolution strategies emphasize behavioral changes at the level of the individual. Genetic algorithms stress chromosomal operators. The evolutionary

process can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. In the case of optimization of noncontinuous and non-smooth function, EAs are much better than nonlinear programming.

### 1) Evolutionary Programming (EP)

EP uses probability transition rules to select generations. Each individual competes with other individuals in a combined population of the old generation and the mutated old generation. The competition results are valued using a probabilistic rule. The winners of the old generation constitute the next generation.

Lai et al. [28] gave the procedure of EP for RPP briefly as follows:

1. Initialization: The initial population of control variables is selected randomly from the set of uniformly distributed control variables ranging over their upper and lower limits. The fitness score  $f_i$  is obtained according to the objective function and the environment.
2. Statistics: The maximum fitness  $f_{\max}$ , minimum fitness  $f_{\min}$ , the sum of fitness  $\sum f$ , and average fitness  $f_{\text{avg}}$  of this generation are calculated.
3. Mutation: Each selected parent  $p_i$  is mutated and assigned to  $p_{i+m}$  in accordance with the following equation:

$$p_{i+m,j} = p_{i,j} + N(0, \beta(x_{j\max} - x_{j\min}) \frac{f_i}{f_{\max}}), j = 1, 2, \dots, n$$

where  $n$  is the number of decision variables in an individual;  $p_{i,j}$  denotes the  $j$ th element of the  $i$ th individual;  $N(\mu, \sigma^2)$  represents a Gaussian random variable with mean  $\mu$  and variance  $\sigma^2$ ;  $x_{j\max}$  and  $x_{j\min}$  are the maximum and minimum limits of the  $j$ th element;  $\beta$  is the mutation scale which is given as  $0 < \beta \leq 1$ .

4. Competition: The individuals have to compete with each other for getting the chance to be transcribed to the next generation. A weight value  $W_i$  of the  $i$ th individual is calculated by the following competition:

$$W_i = \sum_{t=1}^N W_{i,t}$$

where  $N$  is the competition number generated randomly;  $W_{i,t}$  is either 0 for loss or 1 for win during the competition procedure. The value of  $W_{i,t}$  is given in the following equation:

$$W_{i,t} = \begin{cases} 1 & U_t < \frac{f_r}{f_r + f_i} \\ 0 & \text{otherwise} \end{cases}$$

where  $f_r$  is the fitness of the randomly selected individual  $p_r$  and  $f_i$  is the fitness of  $p_i$ ;  $U_t$  is randomly selected from a uniform distribution set,  $U(0,1)$ . When all  $2m$  individuals get their competition weights, they

will be ranked in a descending order according to their corresponding value  $W_i$ . The first  $m$  individuals are selected along with their corresponding fitness  $f_i$  to be the basis for the next generation.

5. Convergence test: If the convergence condition is not met, the mutation and competition processes will run again. The maximum generation number can be used for the convergence condition. Generations are also repeated until

$$(f_{\text{avg}} / f_{\max}) \geq \delta$$

where,  $\delta$  should be very close to 1, which represents the degree of satisfaction.

### 2) Evolutionary Strategy

Evolutionary strategy is similar to evolutionary programming, and the difference is as follows [29]:

In the mutation process, each selected parent  $p_i$  is mutated and added to its population following the rule,

$$p_{i+m,j} = p_{i,j} + N(0, \beta \nabla_{\text{dev}}), j = 1, 2, \dots, n$$

where,  $\nabla_{\text{dev}}$  is fixed, and its value depends on the size of decision variables.

### 3) Genetic Algorithms (GAs)

GAs developed by Holland et al. [30] are a probabilistic heuristic approach that finds a solution to satisfy the fitness function, which combines the survivor of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm. GAs emphasize models of DNA selection as observed in nature, such as crossover and mutation, which are applied to abstracted chromosomes. This is in contrast to ES and EP, which emphasize mutational transformations that maintain behavioral linkage between each parent and its offspring.

Lee et al. [29] gave the simple genetic algorithm as follows:

1. Initial population generation: an initial population of binary strings is created randomly. Each of these strings represents one feasible solution to the search problem.
2. Fitness evaluation: The fitness of each candidate is evaluated through some appropriate measure. After the fitness of the entire population has been determined, it must be determined whether or not the termination criterion has been satisfied. If the criterion is not satisfied, then we continue with the three genetic operations of reproduction, crossover, and mutation.
3. Selection and reproduction: This operation yields a new population of strings that reflect the fitness of the previous generation's fit candidates.
4. Crossover: Crossover involves choosing a random position in the two strings and swapping the bits that occur after this position.
5. Mutation: Mutation is performed sparingly, typically after every 100-1000 bit transfers from crossover, and it involves selecting a string at random as well as a bit

position at random and changing it from a 1 to a 0 or vice-versa. It is used to escape from a local minimum. After mutation, the new generation is complete, and the procedure begins again with the fitness evaluation of the population.

Ajjarapu [31] proposed to solve the optimal capacitor placement problem by genetic based algorithms, and concludes that GAs offer great robustness by:

1. Searching for the best solution from a population point of view and not by starting at a single point.
2. Avoiding derivatives and using payoff information (objective function). In order to perform an effective search for better structures, GAs only require objective function values associated with individual strings.
3. Unlike many methods, GAs use probabilistic transition rules to guide their search.

Iba [32] proposed an approach, which is based on GAs, but quite different from conventional GAs. Their principal features are: searches multiple paths to reach a global optimum; uses various objective functions simultaneously; treats integer/discrete variables naturally; uses two unique intentional genetic operations, which are “Interbreeding: Crossover using network topology” and “Manipulation: AI based stochastic ‘if-then’ rules”. Test results show good convergence characteristics and feasible computing speed.

Urdaneta et al. [33] developed a hybrid methodology technique, which is based upon a modified genetic algorithm. It is applied at an upper level stage, and a successive linear program at a lower level stage. The location of the new sources are decided at a higher layer, while the type and size of the sources is decided at a lower layer. This partition is made to take advantage of the fact that at the upper layer the decision problem consists solely of binary variables, representing a combinatorial optimization problem. Hybrid algorithms have been proposed to combine the strengths of the approaches that search for global solutions with the speed of algorithms specifically adapted to the particular characteristics of the problem.

Dong et al. [34] employed several techniques to improve the performance of the GA search capability: mutation probability control, which adjusts the mutation probability according to the generation number and individual’s fitness in the current generation to give search diversity in the beginning and accelerating the search process close to the end. Another technique used is the elitism or best survival technique, which keeps the best fit individual throughout the generations to record the best solutions as well as to accelerate the search process toward the best fit one.

GAs are good methods to obtain global optimization. However, excessive time consumption will limit their applications in power systems, especially during real-time operation. Most proposed methods were designed for online applications, for which convergence reliability, processing speed, and memory storage are of prime concern. Even though some modifications were made to improve the computing time, these methods are usually confined to fixed

VAR compensation locations and may not be suitable for large-scale power systems with strict constraints and serious VAR shortages [35].

### C. Neural Network Approach

An artificial neural network (ANN) can be defined as: a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain. These processing elements are usually organized into a sequence of layers with full or random connections between the layers. This arrangement is shown in Fig. 1. The input layer is a buffer that presents data to the network. The top layer is the output layer, which presents the output response to a given input. The other layer is called the middle or hidden layer because it usually has no connections to the outside world. Generally, ANN has four features as follows [36]:

1. It learns by example. Indeed, one of the most critical characteristics of ANN is the ability to utilize examples taken from data and to organize the information into a form that is useful. From the network’s viewpoint, the process of training the net becomes learning, or adapting to, the training set, and the prescription for how to change the weights at each step is the learning rule. Learning usually is accomplished by modification of the weight values. A system of differential equations for the weight values can be written as

$$\dot{w}_{ij} = G_i(w_{ij}, x_i, x_j, \dots),$$

where  $G_i$  represents the learning law.

2. It constitutes a *distributed, associative* memory. *Distributed* means the information is spread among all of the weights that have been adjusted in the training process. *Associative* means that if the trained network is presented with a partial input, the network will choose the closest match in the memory to that input and generate an output that corresponds to a full input.
3. It is fault-tolerant. Even when a large number of the weights are destroyed and the performance of the neural network degrades, the system does not fail catastrophically because the information is not contained in just one place but is, instead, distributed throughout the network.
4. It is capable of pattern recognition. Neural networks are good pattern recognizers, even when the information comprising the patterns is noisy, sparse, or incomplete.

An optimization neural network (ONN) can be used to solve mathematics programming problems and has attracted much attention in recent years. Kennedy et al. [37] presented a modified neural network model, which can be used to solve nonlinear programming problems. However, there were some

feasibility problems. Recently, Maa et al. [38] proposed a two-phase optimization neural network in which the stability and feasibility of the neural networks have been modified to some extent.

Zhu et al. [35] developed a new ONN approach to study the reactive power problem. The ONN changes the solution of the optimization problem into an equilibrium point or equilibrium state of a nonlinear dynamic system, and changes optimal criterion into energy functions for a dynamic system. Due to its parallel computational structure and the evolution of dynamics, the ONN approach is superior to traditional optimization methods. Abdul-Rahman et al. [39] integrated fuzzy sets, artificial neural networks, and expert systems into the VAR control problem and included the load uncertainty. The ANN, enhanced by fuzzy sets, is used to determine the memberships of control variables corresponding to the given load values. A power flow solution determines the corresponding state of the system.

The learning capability of ANN spurred a surge of interest in employing artificial intelligence (AI) for the on-line solution of different power system problems. Furthermore, any modeling deficiency in applying algorithmic or rule-based approaches to power systems may cause the corresponding approach to deteriorate. However, failures of some neurons in the ANN may only degrade its performance, it may recover completely from such failures with additional training.

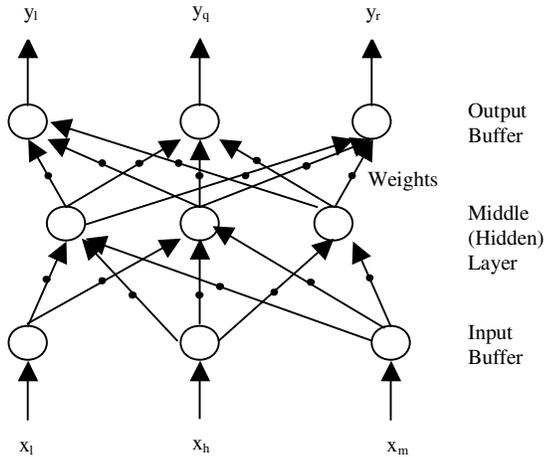


Fig. 1. Example of a neural network architecture.

#### D. Index Method

Another way for ranking and selecting VAR source locations is to use different indices. Some only use one index to decide the location, and others combine several indices with different weights. The following is an introduction to several different index methods.

##### 1) Sensitivity Analysis

Negnevitsky et al. [40] use stability performance index ( $SI$ ):

$$SI_i = \sum_{j=1}^{N_g} S_{ji}$$

where  $N_g$  is the number of generators;  $S_{ji}$  is the  $j$ th element in the  $i$ th column of the sensitivity matrix  $S$ .  $S$  is the sensitivity matrix of the generator reactive power with respect to the reactive power compensation, which is defined by  $\Delta Q_G/\Delta Q_L$ , where  $\Delta Q_G$  is the change in the generator bus reactive power;  $\Delta Q_L$  is the change in the load bus reactive power. Because the most effective location of a reactive power compensator is where the effect of the compensator's operation in the reactive power reserve of generators is strongest, the higher the value of  $SI$ , then the better its stability control capability. Gribik et al. [41] calculated the sensitivities of system losses to real power load at all buses of the network for ranking the prospective VAR compensation location.

The above two papers are examples of using only one sensitivity index, and the following papers all have several sensitivity indices. Ajjarapu et al. [42] used bus sensitivity, branch sensitivity, and generator sensitivity to measure the closeness of reaching the steady state voltage stability limits. Refaey et al. [43] proposed a technique for combining three indices named the voltage performance index (VPI), the steady-state stability index, and the real power loss index, which are given different weights depending on their values.

##### 2) Cost Benefit Analysis (CBA)

In the CBA scheme, all load buses have been considered as possible locations for installation of capacitors, and CBA is carried out to determine which of the capacitors are cost-effective and in which size. Chattopadhyay et al. [44] carried out the CBA at the first stage and then incorporated the outcome of CBA in the OPF computation. This approach is obviously superior to the traditional one in which the sites of new VAR sources are either simply estimated or directly assumed. However, it neglects the effect of voltage improvement and system loss decrease in the selected VAR source sites.

##### 3) Voltage Stability Margin Method (VSMM)

In the new competitive environment, the system operator should consider voltage, security, and transmission loss at the same time. The benefit-to-cost ratio index and sensitivity index reflect the economy and sensitivity in the selection of VAR source sites in the normal case, but they can not reflect the voltage security or stability under the contingency cases. Momoh and Zhu [45] presented a voltage security margin method (VSMM) to calculate the bus voltage security margin index in the selection of VAR source sites, which is defined as

$$VSMM'_i = \frac{V'_i(0) - \min[V'_i(J)]}{V'_i(0) - V_{i\min}}$$

where

$V'_i(0)$ : voltage magnitude at bus  $i$  at time  $t$  in the normal case;

$V_i^t(l)$  : voltage magnitude at bus  $i$  at time  $t$  in the case of line  $l$  outage;

$V_{i\min}$  : lower limit of voltage at bus  $i$ ;

$VSMI_i^t$  : hourly-based voltage security margin index.

#### 4) Analytic Hierarchical Process (AHP)

When several indices are used for one problem, the bus ranking results are not necessarily the same. Unfortunately, it is difficult to find a unified process for ranking these results. Zhu et al. [35] proposed an analytic hierarchical process, which can comprehensively consider the effect of several independent indices and make a unified decision according to various judgment matrices.

### IV. CONCLUSIONS

The analysis of RPP is one of the most complex problems in power systems, as it requires the simultaneous minimization of two objective functions. The first one deals with the minimization of operational costs by reducing real power loss and improving the voltage profile. The second objective minimizes the allocation costs of additional reactive power sources. In fact it has a very complicated, partially discrete, partially continuous formulation with a non-differentiable nonlinear objective function. In this paper, nine category methods are introduced, and their advantages and shortcoming are compared with each other. Table I shows the comparison result of these methods' characteristics.

TABLE I. COMPARISON OF METHODS' CHARACTERISTICS

Characteristics	Local Convergence	Global Convergence	Local Optimum	Global Optimum
<b>Method</b>				
GRG		✓	✓	
Newton Method	✓		✓	
SQP		✓	✓	
LP		✓	✓	
MIP		✓	✓	
DWDM		✓	✓	
LRDM		✓	✓	
GBD	✓		✓	
HM	✓		✓	
SA		✓		✓
EAs		✓		✓
ANN		✓		✓

Even the Generalized Reduced Gradient (GRG) Method has global convergence, but its linear convergence rate is very slow. Newton's method has a quadratic convergence rate but suffers from having only local convergence. Successive Quadratic Programming (SQP) can guarantee the quadratic convergence rate and global convergence, which is superior compared with the two former methods. Linear Programming (LP) can support the reliability, but when it handles two objective functions, it is difficult to find an exact

solution. Mixed Integer Programming (MIP) can exactly construct the objective function of the RPP problem, but a decomposition technique has to be employed, which makes the whole algorithm bulky. The Dantzig-Wolfe Decomposition Method (DWDM), Lagrange Relaxation Decomposition Method (LRDM), and Generalized Benders Decomposition (GBD) can all make the complicated problem simple, so as to reduce the number of iterations, computation time, and the memory space. In addition, they use a diversity of approaches in one problem. The binary search in the Heuristic Method (HM) is an educated guess method that can not guarantee a solution.

In the second period, Simulated Annealing (SA), Evolutionary Algorithms (EAs), and Artificial Neural Networks (ANNs) were all shown to be able to obtain a global optimum, and the solution is robust to the starting point. One common trait in this period is the combination of several methods, especially some decomposition method and an artificial intelligent method, but they have the common drawback of time-consuming computation.

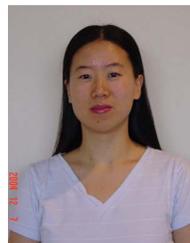
There is still growing interest in the field of VAR planning as evidenced by the recent works reported in the literature.

### REFERENCES

- [1] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear programming theory and algorithms*, John Wiley & Sons, Nov. 1992
- [2] F. F. Wu, G. Gross, J. F. Luini, P. M. Look, "A Two-stage Approach To Solving Large-scale Optimal Power Flows," *Power Industry Computer Applications Conference, 1979. PICA-79. IEEE Conference Proceedings*, May 15-18, 1979, pp. 126-136.
- [3] C. J. Parker, I. F. Morrison, D. Sutanto, "Application of an optimisation method for determining the reactive margin from voltage collapse in reactive power planning," *IEEE Trans. on PAS*, vol. 11, no. 3, Aug. 1996, pp. 1473-1481.
- [4] O. Crisan, M. A. Mohtadi, "Efficient identification of binding inequality constraints in optimal power flow Newton approach," *Generation, Transmission and Distribution, IEE Proceedings C*, vol. 139, no. 5, Sept. 1992, pp. 365-370.
- [5] A. D. Papalexopoulos, C. F. Imparato, F. F. Wu, "Large-scale optimal power flow: effects of initialization, decoupling and discretization," *IEEE Trans. on PAS*, vol. 4, no. 2, May 1989, pp. 748-759.
- [6] B. Kermanshahi, K. Takahashi, Y. Zhou, "Optimal operation and allocation of reactive power resource considering static voltage stability," *1998. Proceedings. POWERCON '98. 1998 International Conference on Power System Technology*, Vol. 2, 18-21 Aug. 1998, pp. 1473-1477.
- [7] N. S. Rau, "Radial equivalents to map networks to market formats - an approach using quadratic programming," *IEEE Trans. on PAS*, vol. 16, no. 4, Nov. 2001, pp. 856-861.
- [8] K. R. C. Mamandur, R. D. Chenoweth, "Optimal control of reactive power flow for improvements in voltage profiles and for power loss minimization," *IEEE Trans. on PAS*, vol. 100, no. 7, July 1981, pp. 3185-3193.
- [9] A. Venkataramana, J. Carr, R. S. Ramshaw, "Optimal reactive power allocation," *IEEE Trans. on PAS*, vol. PWRS-2, no.1, Feb. 1987, pp. 138-144.
- [10] K. Iba, H. Suzuki, K.-I. Suzuki, K. Suzuki, "Practical reactive power allocation/operation planning using successive linear programming," *IEEE Trans. on PAS*, vol. 3, no. 2, May 1988, pp.558-566.
- [11] M. Yehia, R. Ramadan, Z. El-Tawail, K. Tarhini, "An integrated technico-economical methodology for solving reactive power compensation problem," *IEEE Trans. on PAS*, vol. 13, no. 1, Feb. 1998, pp. 54-59.

- [12] W. R. Thomas, A. M. Dixon, D. T. Y. Cheng, R. M. Dunnett, G. Schaff, J. D. Thorp, "Optimal reactive planning with security constraints," *IEEE Power Industry Computer Application Conference*, 7-12 May 1995, pp. 79–84.
- [13] B. Venkatesh, G. Sadasivam, M.A. Khan, "Optimal reactive power planning against voltage collapse using the successive multiobjective fuzzy LP technique," *IEE Proceedings on Generation, Transmission and Distribution*, vol. 146, no. 4, July 1999, pp. 343–348.
- [14] K. Aoki, M. Fan, A. Nishikori, "Optimal VAr planning by approximation method for recursive mixed-integer linear programming," *IEEE Trans. on PAS*, vol. 3, no. 4, Nov. 1988, pp. 1741–1747.
- [15] M. S. Bazaraa, J. J. Jarvis, H. D. Sherali, *Linear Programming and NetworkFlows*, John Wiley & Sons, 1990.
- [16] N. Deeb, S. M. Shahidehpour, "Linear reactive power optimization in a large power network using the decomposition approach," *IEEE Trans. on PAS*, vol. 5, no. 2, May 1990, pp. 428–438.
- [17] A. M. Geoffrion, "Lagrangian relaxation for integer programming," *Mathematical ProgrammingStudy*, vol. 2, 1974, pp. 82–113.
- [18] T. Gomez, I. J. Perez-Arriaga, J. Lumbreras, V. M. Parra, "A security-constrained decomposition approach to optimal reactive power planning," *IEEE Trans. on PAS*, vol. 6, no. 3, Aug. 1991, pp. 1069–1076.
- [19] Y.-Y. Hong, D. I. Sun, S.-Y. Lin, C.-J. Lin, "Multi-year multi-case optimal VAR planning," *IEEE Trans. on PAS*, vol. 5, no. 4, Nov. 1990, pp. 1294–1301.
- [20] K. H. Abdul-Rahman, S. M. Shahidehpour, "Application of fuzzy sets to optimal reactive power planning with security constraints," *Power Industry Computer Application Conference*, 4-7 May 1993, pp. 124–130.
- [21] N. Yorino, E. E. El-Araby, H. Sasaki, S. Harada, "A new formulation for FACTS allocation for security enhancement against voltage collapse," *IEEE Trans. on PAS*, vol. 18, no. 1, Feb. 2003, pp. 3–10.
- [22] N. I. Deeb, S. M. Shahidehpour, "Cross decomposition for multi-area optimal reactive power planning," *IEEE Trans. on PAS*, vol. 8, no. 4, Nov. 1993, pp. 1539–1544.
- [23] J. R. S. Mantovani, A. V. Garcia, "A heuristic method for reactive power planning," *IEEE Trans. on PAS*, vol. 11, no. 1, Feb. 1996, pp. 68–74.
- [24] R. V. V. Vidal, *Applied Simulated Annealing*, Springer-Verlag, 1993.
- [25] Y.-T. Hsiao, C.-C. Liu, H.-D. Chiang, Y.-L. Chen, "A new approach for optimal VAr sources planning in large scale electric power systems," *IEEE Trans. on PAS*, vol. 8, no. 3, Aug. 1993, pp. 988–996.
- [26] Y.-T. Hsiao, H.-D. Chiang, C.-C. Liu, Y.-L. Chen, "A computer package for optimal multi-objective VAr planning in large scale power systems," *IEEE Trans. on PAS*, vol. 9, no. 2, May 1994, pp. 668–676.
- [27] W.-S. Jwo, C.-W. Liu, C.-C. Liu, Y.-T. Hsiao, "Hybrid expert system and simulated annealing approach to optimal reactive power planning," *IEE Proceedings on Generation, Transmission and Distribution*, vol. 142, no. 4, July 1995, pp. 381–385.
- [28] L. L. Lai, J. T. Ma, "Application of evolutionary programming to reactive power planning-comparison with nonlinear programming approach," *IEEE Trans. on PAS*, vol. 12, no. 1, Feb. 1997, pp. 198–206.
- [29] K. Y. Lee, F. F. Yang, "Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming," *IEEE Trans. on PAS*, vol. 13, no. 1, Feb. 1998, pp. 101–108.
- [30] J. H. Holland, "Adaptation in natural and artificial system," *The University of Michigan Press*, 1975.
- [31] V. Ajjarapu, Z. Albanna, "Application of genetic based algorithms to optimal capacitor placement," *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems*, 23-26 July 1991, pp. 251 – 255.
- [32] K. Iba, "Reactive power optimization by genetic algorithm," *Power Industry Computer Application Conference*, 4-7 May 1993, pp. 195–201.
- [33] A. J. Urdaneta, J. F. Gomez, E. Sorrentino, L. Flores, R. Diaz, "A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming," *IEEE Trans. on PAS*, vol. 14, no. 4, Nov. 1999, pp. 1292–1298.
- [34] Z.-Y. Dong, D. J. Hill, "Power system reactive scheduling within electricity markets," *International Conference on Advances in Power System Control, Operation and Management*, vol. 1, 30 Oct. – 1 Nov. 2000, pp. 70–75.
- [35] J. Z. Zhu, C. S. Chang, W. Yan, G. Y. Xu, "Reactive power optimisation using an analytic hierarchical process and a nonlinear optimisation neural network approach," *IEE Proceedings – Generation, Transmission, and Distribution*, V=vol. 145, no. 1, Jan. 1998, pp. 89 – 97.
- [36] L. H. Tsoukalas, R. E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, 1997.
- [37] M. P. Kennedy, L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. on Circuits and Systems*, vol. 35, no. 5, May 1988, pp. 554–562.
- [38] C. Y. Maa, M. A. Shanblatt, "A two-phase optimization neural network," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, 1992, pp. 1003–1009.
- [39] K. H. Abdul-Rahman, S. M. Shahidehpour, M. Daneshdoost, "AI approach to optimal VAr control with fuzzy reactive loads," *IEEE Trans. on PAS*, vol. 10, no. 1, Feb. 1995, pp. 88–97.
- [40] M. Negnevitsky, R. L. Le, M. Piekutowsky, "Voltage collapse: case studies," *Power Quality*, 1998, pp. 7–12.
- [41] P. R. Gribik, D. Shirmohammadi, S. Hao, C. L. Thomas, "Optimal power flow sensitivity analysis," *IEEE Trans. on PAS*, vol. 5, no. 3, Aug. 1990, pp. 969–976.
- [42] V. Ajjarapu, P. L. Lau, S. Battula, "An optimal reactive power planning strategy against voltage collapse," *IEEE Trans. on PAS*, vol. 9, no. 2, May 1994, pp. 906–917.
- [43] W. M. Refaey, A. A. Ghandakly, M. Azzoz, I. Khalifa, O. Abdalla, "A systematic sensitivity approach for optimal reactive power planning," *North American Power Symposium*, 15-16 Oct. 1990, pp. 283–292.
- [44] D. Chattopadhyay, K. Bhattacharya, J. Parikh, "Optimal reactive power planning and its spot-pricing: an integrated approach," *IEEE Trans. on PAS*, vol. 10, no. 4, Nov. 1995, pp. 2014–2020.
- [45] J. A. Momoh, J. Zhu, "A new approach to VAr pricing and control in the competitive environment," *Hawaii International Conference on System Sciences*, vol. 3, 6-9 Jan. 1998, pp. 104–111.

## BIOGRAPHIES



**Wenjuan Zhang** (S 2003) received the B.E. in electrical engineering from Hebei University of Technology, China, in 1999 and an M.S. in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003. She worked at the Beijing Coherence Prudence Development Company for one year on power distribution related projects.

She presently is a Ph.D. student in Electrical Engineering at The University of Tennessee. Her research interests are power quality and reactive power planning.



**Leon M. Tolbert** (S 1989 – M 1991 – SM 1998) received the B.E.E., M.S., and Ph.D. in Electrical Engineering from the Georgia Institute of Technology, Atlanta, Georgia

Since 1991, he worked on several electrical distribution projects at the three U.S. Department of Energy plants in Oak Ridge, Tennessee. In 1999, he was appointed as an assistant professor in the Department of Electrical and Computer Engineering at the University of Tennessee, Knoxville. He is an adjunct participant at the Oak Ridge National Laboratory. He does research in the areas of electric power conversion for distributed energy sources, reactive power compensation, multilevel converters, hybrid electric vehicles, and application of SiC power electronics.