Ant Colony System-Based Algorithm for Constrained Load Flow Problem

John G. Vlachogiannis, Nikos D. Hatziargyriou, *Senior Member, IEEE*, and Kwang Y. Lee, *Fellow, IEEE*

***Abstract*—This paper presents the ant colony system (ACS) method for network-constrained optimization problems. The developed ACS algorithm formulates the constrained load flow (CLF) problem as a combinatorial optimization problem. It is a distributed algorithm composed of a set of cooperating artificial agents, called ants, that cooperate among them to find an optimum solution of the CLF problem. A pheromone matrix that plays the role of global memory provides the cooperation between ants. The study consists of mapping the solution space, expressed by an objective function of the CLF on the space of control variables [ant system (AS)-graph], that ants walk. The ACS algorithm is applied to the IEEE 14-bus system and the IEEE 136-bus system. The results are compared with those given by the probabilistic CLF and the reinforcement learning (RL) methods, demonstrating the superiority and flexibility of the ACS algorithm. Moreover, the ACS algorithm is applied to the reactive power control problem for the IEEE 14-bus system in order to minimize real power losses subject to operating constraints over the whole planning period.**

***Index Terms*—Ant colony system (ACS), combinatorial optimization, constrained load flow (CLF), reinforcement learning (RL).**

I. Introduction

Many optimization problems in power systems can be expressed as combinatorial optimization problems, such as the constrained load flow (CLF) problem. The CLF problem deals with the offline adjustment of the power system control variables in order to satisfy physical and operating constraints. A number of traditional algorithms have been developed in order to solve this problem [1]–[8]. These algorithms are based on the modification of the Jacobian matrix formed in the standard load flow method, using sensitivity or injection-changing-error feedback control [1], [2] or evolutionary computation techniques [3], [4]. The CLF problem is also expressed as a constrained optimization problem falling within the general class of optimal power flow (OPF) problems [5]–[8].

These methods, however, are inefficient in providing offline settings of control variables that must remain optimal for a whole planning period. The problem of the offline control settings has been tackled by the probabilistic CLF formulation [9]. The method [9] takes into account load uncertainties and generating unit unavailability modeled as probability density functions. This method provides near-optimum offline control settings. Recent research solves the CLF problem by means of the heuristic reinforcement learning (RL) method [10]–[12]. The RL modified the CLF problem as a combinatorial optimization problem [13]. In this, optimal control settings are learned by experience adjusting a closed-loop control rule, which maps operating states to control actions by means of reward values [13].

In this paper, the CLF problem is solved by means of the heuristic ant colony system (ACS) method. Dorigo has proposed the first ACS in his Ph.D. dissertation [14]. The ACS method belongs to biologically inspired heuristics (meta-heuristics) methods. Real ants are capable of finding the shortest path from the food source to their nest, without using visual cues, but by exploiting pheromone information. While walking, real ants deposit pheromone trails on the ground and follow pheromones previously deposited by other ants. This behavior has inspired the ACS algorithm in which a set of artificial ants cooperate in solving a problem by exchanging information via pheromones deposited on a graph. Currently, most works have been done in the direction of applying ACS to the combinatorial optimization problems [15]–[18]. For most of these applications, the results show that the ACS-based approach can outperform other heuristic methods. In power systems, the ACS has been applied to solve the optimum generation scheduling problems [19], [20] and the optimum switch relocation and network reconfiguration problems for distribution systems [21], [22]. It is rather difficult to find a single search space, configuration, and a parameter set of an ACS algorithm that can satisfy every optimization problem. Therefore, there is a need for the development of an improved version of the ACS algorithms [14]–[21] tailored to solve the CLF problem. The ACS algorithm proposed in this paper formulates the CLF problem as a combinatorial optimization problem. As an example, the settings of control variables (tap-settings, VAr compensation blocks, etc.) are combined in order to achieve optimum voltage values at the nodes of a power system. In our approach, the graph that describes the settings of control variables of the CLF problem is mapped on the ant system (AS)-graph, which is the space that the artificial ants will walk. Specifically, the objective function of the ACS algorithm has similar formulation with the Q-learning [13] reward function. The objective function "fires" the transition function, which gives the probability for an ant to select an edge to walk. In this paper, for computational simplicity, the transition function considers only the trail intensity for the transition probability [14], [21], i.e., the trail that more ants choose will have more probability to be selected.

Manuscript received February 9, 2004; revised January 26, 2005. Paper no. TPWRS-00060-2004.

J. G. Vlachogiannis is with the Industrial and Energy Informatics (IEI) Laboratory, R.S. Lianokladiou, 35100 Lamia, Greece (e-mail: vlachogiannis@usa.com).

N. D. Hatziargyriou is with the Department of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece (e-mail: nh@power.ece.ntua.gr).

K. Y. Lee is with the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: kwanglee@psu.edu).

Digital Object Identifier 10.1109/TPWRS.2005.851969

The paper is organized in six sections. Section II formulates the constrained load flow problem, and Section III describes the basic concepts of the ACS. In Section IV, the ACS algorithm is implemented to the CLF problem. In Section V, the results obtained by the application of the ACS algorithm to the IEEE 14-bus system and to the IEEE 136-bus system are presented and discussed. The ACS algorithm is compared with a conventional probabilistic method and the reinforcement learning recently introduced by Ernst *et al.* in power system control [26], [27]. Specifically, the results are compared with those obtained by the probabilistic CLF [9] and the Q-learning method [13], showing the superiority of the proposed ACS algorithm. Moreover, the ACS algorithm is applied to the reactive power control problem for the IEEE 14-bus system in order to minimize real power losses while simultaneously satisfying operating constraints over the whole planning period. Finally, in Section VI, general conclusions are drawn.

II. CLF Problem

The load flow problem can be expressed by the next two sets of nonlinear equations:

$$\begin{aligned} Y &= g(X,U) \\ Z &= h(X,U) \end{aligned} \tag{1}$$

where

Y vector of nodal power injections

Z vector of constrained variables (power flows, reactive powers of PV buses, etc.);

X state vector (voltage angles and magnitudes);

U control vector (transformer tap settings, shunt compensation, voltage and power at PV buses, etc.).

The objective of the constrained load flow is to maintain some or all elements of X and Z vectors within given operating limits under the uncertainty of generating units' availabilities and load uncertainties. This can be achieved by selecting appropriate (robust) values of control variables under random variations of loads and generations (noise factors) within their operating range. The next sections of this paper propose a technique to maintain constrained variables within operating limits over the whole planning period using the ACS algorithm.

III. Basic Concepts of ACS

ACS algorithms simulate the behavior of real ants [14], [15]. They are based on the principle that using simple communication mechanisms, an ant group is able to find the shortest path between any two points. During their trips, a chemical trail (pheromone) is left on the ground. The pheromone guides other ants toward the target point. For one ant, the path is chosen according to the quantity of pheromone. The pheromone evaporates over time (i.e., it loses quantity if other ants lay down no more pheromone). If many ants choose a certain path and lay down pheromones, the quantity of the trail increases, and thus, this trail attracts more and more ants. The artificial ants (or simply, ants) simulate the transitions from one point r to another point s, according to the improved version of ACS, namely the Max-Min AS (MMAS) algorithm [23], as follows.

If the ant k is at point r, has the next point been visited? The ant k maintains a tabu list N_r^k in memory that defines the set of points still to be visited when it is at point r. The ant k chooses to go from point r to point s during a tour with a probability given by [23]

$$p(r,s) = \frac{\gamma(r,s)}{\sum\limits_{l} \gamma(r,\ell)} \quad \mathrm{s}, \ell \in N_r^k \tag{2}$$

where matrix $\gamma(r,s)$ represents the amount of the pheromone trail (pheromone intensity) between points r and s.

Then, the pheromone trail on coupling (r,s) is updated according to

$$\gamma(r,s) = \alpha \cdot \gamma(r,s) + \Delta\gamma^k(\mathrm{r,s}) \tag{3}$$

where α with $0 < \alpha < 1$ is the persistence of the pheromone trail, so that $(1-\alpha)$ to represent the evaporation and $\Delta\gamma^k(r,s)$ is the amount of pheromone that ant k puts on the trail (r,s). The pheromone update $\Delta\gamma^k(r,s)$ reflects the desirability of the trail (r,s), such as shorter distance, better performance, etc., depending on the application problem. Since the best tour is unknown initially, an ant needs to select a trail randomly and deposits pheromone in the trail, where the amount of pheromone will depend upon the pheromone update rule (3). The randomness implies that pheromone is deposited in all possible trails, not just in the best trail. The trail with favorable update, however, increases the pheromone intensity more than other trails.

After all ants have completed their tours, global pheromone is updated in the trails of the ant with the best tour executed. In the next section, the MMAS algorithm is extended and modified to solve the CLF problem.

IV. Development of Ant Colony System for the CLF Problem

The settings of control variables (tap-settings, VAr compensation blocks, etc.) are combined in order to achieve the power system constraints. In our approach, the graph that describes the settings of control variables of the CLF problem is mapped on the AS-graph, which is the space that the artificial ants will walk. Fig. 1 shows the AS-graph (searching space) for the CLF problem. All possible candidate discrete settings for a control variable are represented by the states r of the AS-graph ($r = 1,\ldots,m$). The control variables are represented by the stages i ($i = 1,\ldots,n$), where n is the number of the control variables. Each ant will start its tour at the home colony and stop at the destination. The ACS algorithm proceeds as follows.

An operating point comprising a load and generation pattern (operating point of the whole planning period of the system) is randomly created. For this operating point, first of all, the AS graph is created, and all paths receive an amount of pheromone that corresponds to an estimation of the best solution so that ants test all paths in the initial iterations. Therefore, the ACS-algorithm achieves the best exploration of the AS-graph in the earlier iterations of convergence and better exploitation at the latest.

Then, ant k chooses the next states to go to in accordance with the transition probability calculated by (2). When the ant k moves from one stage to the next, the state of each stage will

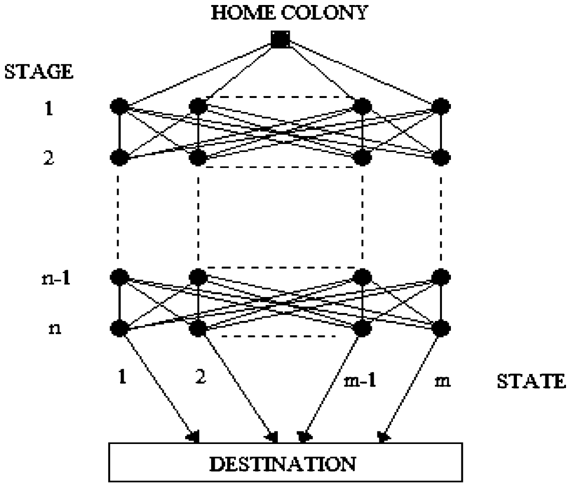

Fig. 1. Search space for the CLF problem.

be recorded in a location list J^k. After the tour of ant k is completed, its location list is used to compute its current solution. Then, the pheromone trails composed by nodes of location list J^k are updated in accordance with (3) (local update). For the purpose of this research, the pheromone update $\Delta\gamma^k(r,s)$ is chosen as

$$\Delta\gamma^k(r,s) = \frac{1}{Q \cdot f} \tag{4}$$

where f is the objective function, and Q is a large positive constant.

Application of the ACS algorithm to the CLF problem is linked to the choice of an objective function f, such as the limits of the constrained variables to be satisfied for the whole planning period. An enforced empirical strategy is to consider the variations of constrained variables close to the means of their operating intervals. The objective function f is computed by the average of all constrained variables, normalized in the interval [0, 1], as follows:

$$f = \frac{1}{n} \cdot \sum_{j=1}^{n} \left| \frac{2z_j - z_{j\max} - z_{j\min}}{z_{j\max} - z_{j\min}} \right| \tag{5}$$

where n expresses the number of constrained variables, z_j the value of jth constrained variable, and $(z_{j\min}, z_{j\max})$ are its lower and upper limits, respectively.

It must be noticed that $f = -r$, where r is the immediate rewards used by the Q-learning algorithm to the CLF problem [13]. The objective function f has this formulation in order for the results provided by the ACS and the Q-learning algorithms to be compared.

In order to exploit the iteration in finding the best solution, the next two steps are considered.

a) When all ants complete their tours, load flow is run, and the objective function (5) is calculated for each run. Then, the pheromone trails (r,s) of the best ant tour [ant with minimum objective function (5)] is updated (global update) as

$$\gamma(r,s) = \alpha \cdot \gamma(r,s) + \frac{R}{f_{\text{best}}} \quad r,s \in J_{\text{best}}^k \tag{6}$$

where R is a large positive constant. Both Q in (4) and R are arbitrarily large numbers. Empirical tests have shown that the ACS-algorithm converges faster when Q is almost equal to R.

b) To avoid search stagnation (the situation where all the ants follow the same path, that is, they construct the same solution [15]), the allowed range of the pheromone trail strengths is limited to

$$\gamma(r,s) = \left\{ \begin{array}{ll} \tau_{\min} & \text{if } \gamma(r,s) \le \tau_{\min} \\ \tau_{\max} & \text{if } \gamma(r,s) \ge \tau_{\max} \end{array} \right\}. \tag{7}$$

For our study, the limits are chosen as

$$\tau_{\max} = \frac{1}{\alpha \cdot f_{\text{gbest}}} \tag{8}$$

where f_{gbest} is the global best solution (best over the whole past iterations), and

$$\tau_{\min} = \frac{\tau_{\max}}{M^2} \tag{9}$$

where M is the number of ants.

The procedure is repeated for a large number of operating states covering the whole planning period. Once we have the set of optimal control settings for a large number of operating points, the one that minimizes the sum of multiobjective function (mtf) over the whole planning period is defined as a greedy-optimum control setting

$$\begin{aligned} \text{mtf} &= \min\{\text{total of objective functions}\} \\ &= \min\left(\sum_{\text{over whole planning period}} f \right). \end{aligned} \tag{10}$$

Table I shows the execution steps of the ACS algorithm applied to the CLF problem.

V. Results

The ACS algorithm is applied to adjust reactive control variables in the IEEE 14-bus test system shown in Fig. 2. The test system consists of the slack bus (node 1), three PV (nodes 2, 3, and 6), ten PQ buses, and 20 branches. It has been used in many probabilistic studies. The network data and load probabilistic data are the same as used in [9]. They comprise six discrete distributions for the active load (at nodes 3, 6, 9, 10, 11, and 14), four discrete distributions for the reactive load (at nodes 9, 10, 11, and 14), with three to five impulses each and eight normal distributions for active and reactive loads at the remaining buses. The total installed capacity is equal to 4.9 p.u. and comprises 14 capacitor banks at node 1, four banks at node 2, two banks at node 3, and two banks at node 6. The voltage at all PV buses is taken equal to 1.0 p.u. and the slack bus voltage equal to 1.02 p.u. A fixed network topology is assumed. The control variables comprise all transformer taps (t) and reactive compensation (b) at bus 9 (see Fig. 2).

The upper part of Table II shows the limits of the control variables Umin and Umax and the discrete steps in variation. The transformer taps ($t56$, $t49$, and $t47$) are in 16 steps, while the reactive compensation ($b9$) is in nine steps. Therefore, the last step

TABLE I
ACS ALGORITHM TO THE CLF PROBLEM

1. Create the AS-graph (search space) that represents the discrete settings (states) of the control variables (stages).

2. Insert the pheromone matrix $\gamma(m,n)$ according to nodes of AS-graph, where n is the number of stages and m the number of states.

3. Initialize the pheromone matrix $\gamma(m,n) = \gamma_0(m,n) = \tau_{\max}$ (in (8), in this case f_{gbest} is an initial estimation of the best solution).

4. Repeat for a given number of operating points over the whole planning period.

4.1 Repeat until the system convergence or iteration is less than a given maximum number.

4.1.1 Place randomly M ants on the states of the 1st stage (i = 1).

4.1.2 For k=1 to M

4.1.2.1 For i = 2 to n

4.1.2.1.1 When the ant-k has selected the r-state of the (i-1)-stage, it currently chooses the s-state of the (i)-stage in which will move according to transition rule (2).

4.1.2.1.2 Move the ant-k to s-state of i–stage.

4.1.2.1.3 Record s to J^k, and set $r = s$.

4.1.3 Run power flow for each ant.

4.1.4 Calculate the objective function (5) for each ant.

4.1.5 Update the pheromone of (r,s)-trails for each ant, using the local pheromone update formulae (3), (4).

4.1.6 Update the pheromone of (r,s)-trails belonging to best ant tour (f_{best}), using the pheromone update formula (6).

4.1.7 In order to avoid the ants stagnations, enforce the limits (7)-(9).

4.2 Enforce each of the best control settings over the whole planning period and calculate (10).

5. Choose as a greedy-optimum control setting the one that minimizes (10).

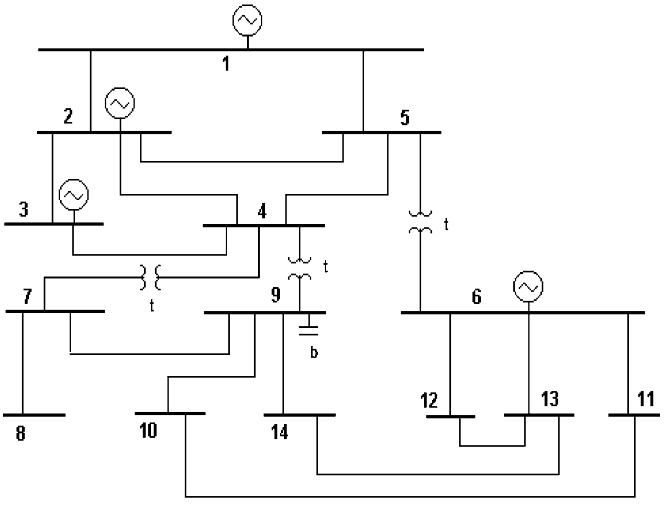

Fig. 2. IEEE 14-bus system.

of $b9$ ($b9 = 0.24$) is repeated for the next seven steps, for steps 10 through 16. This makes the pheromone matrix $\gamma(m,n)$ of the AS-graph well defined for all stages and states. In the lower part of Table II, the upper and lower limits of all constrained variables Wmax and Wmin are shown.

In our study, the following ACS parameters (see Table I) are chosen: $M = 100$, $n = 4$, $m = 16$, and $Q = R = 1\,000\,000$, and the initial best solution is estimated at 0.0001. The parameter α from our experience shows that any value in the range [0.88 0.999] works well. In this paper, it is chosen as $\alpha = 0.9865$. In this study, the search will be terminated if one of the following criteria is satisfied: a) The number of iterations since the last change of the best solution is greater than 1000 iterations, or b) the number of iterations reaches 3000 iterations.

TABLE II
LIMITS AND DISCRETIZATION OF ACTIONS AND LIMITS OF CONSTRAINED VARIABLES

Control Actions	Umin	Umax	Step
$t56$	0.90	1.05	0.01
$t49$	0.90	1.05	0.01
$t47$	0.90	1.05	0.01
$b9$	0.00	0.24	0.03

Constrained Variables	Wmin	Wmax
Qg2	0.00	0.30
Qg3	0.00	0.70
Qg6	0.00	0.45
T23	0.00	0.75
T56	0.00	0.50
V4	0.96	1.05
V5	0.96	1.05
V7	0.96	1.05
V8	0.96	1.05
V9	0.96	1.05
V10	0.96	1.05
V11	0.96	1.05
V12	0.96	1.05
V13	0.96	1.05
V14	0.96	1.05

The ACS algorithm can be implemented in a large number of load combinations (operating points) selected over the whole planning period.

In our study, the algorithm learns the optimum control settings for each of 41 operating points selected from the whole planning period. These full correlated operating points are sampled uniformly from the curves of normal and discrete distribution probabilities as follows:

$$\text{Load step} = \left(\mu \pm k \times \frac{3\sigma}{20}\right) \cdot 100\% \tag{11}$$

where $k = 0, 1, 2, 3, \ldots, 20$. The μ and σ are the average values and the standard deviations for normal distributions of loads given in [24]. The μ and σ for the discrete distributions of loads are calculated using the formulae given by [25].

Performance of the ACS algorithm is shown in Figs. 3(a), 3(b), and (3c) depicting the obtained values of objective function (5) during the ACS procedure for the nominal, heavy, and light load, respectively.

The nominal load corresponds to the mean values of the load. The heavy and light load correspond to the 1% confidence limit that all load values are lower and higher than these values, respectively. Figs. 3(a)–3(c) show the convergence of ACS algorithm in a minimum value of (5), achieving the optimum control settings for each of the three operating points corresponding to the heavy, light, and nominal load. Among the 41 optimal control settings, the *greedy-optimum* control settings are those that provide the minimum total function (10). Tests have shown that the calculation of (10) in the above three operating points is sufficient to provide the mtf. Results on the IEEE 14-bus system show that the greedy-optimum control settings that achieve the mtf (10) over the whole planning period are the optimal control settings obtained for the nominal load [see Fig. 3(a)]. In this case, convergence of the ACS algorithm took 1730 iterations. Table III shows the mtf (10) is calculated at 0.732, the greedy-optimum control settings and the operating space of constrained variables, when these

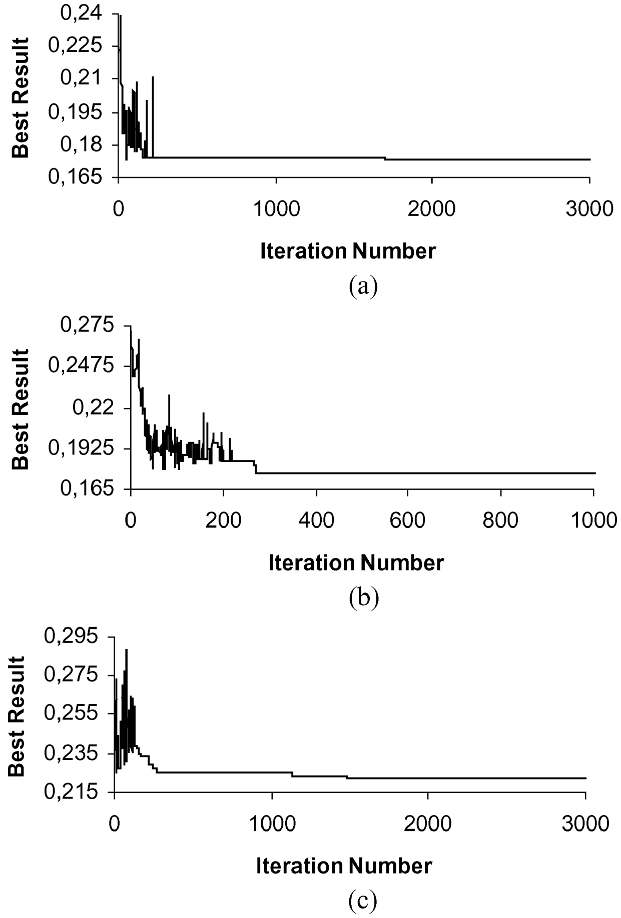

Fig. 3. Performance of ACS algorithm in the (a) nominal, (b) heavy, and (c) light loads.

TABLE III
COMPARISON OF RESULTS BETWEEN ACS, RL, AND PROBABILISTIC LOAD FLOW ON THE IEEE 14-BUS SYSTEM

	ACS Algorithm		Q-Learning Algorithm		Probabilistic Load Flow	
	mtf = 0.732		*mtr* = -0.804		-	
Contr Varib.	Optimal Settings		Greedy-Optimal Settings (a*)		Optimal Settings	
*t*56	1.01		1.03		0.94	
*t*49	0.91		0.97		0.97	
*t*47	0.99		0.90		0.98	
*b*9	0.18		0.12		0.12	
Constr. Varib.	Wmin	Wmax	Wmin	Wmax	Wmin	Wmax
Qg2	-0.5508*	-0.0795*	-0.5250*	-0.0527*	0.2069	0.3160*
Qg3	0.1554	0.5978	0.1751	0.6183	0.6420	0.6812
Qg6	0.0688	0.3606	0.0891	0.3787	0.3065	0.4161
T23	0.2273	0.6847	0.2284	0.6868	0.7223	0.7799*
T56	0.0731	0.2920	0.1675	0.4528	0.4144	0.4931
V4	0.9777	0.9984	0.9742	0.9951	0.9654	0.9731
V5	0.9876	1.0042	0.9864	1.0030	0.9682	0.9744
V7	0.9864	1.0235	0.9901	1.0274	0.9710	0.9857
V8	0.9864	1.0235	0.9901	1.0274	0.9710	0.9833
V9	0.9871	1.0316	0.9840	1.0284	0.9656	0.9833
V10	0.9819	1.0237	0.9775	1.0193	0.9644	0.9803
V11	0.9903	1.0139	0.9831	1.0067	0.9828	0.9912
V12	0.9899	1.0011	0.9802	0.9915	0.9907	0.9936
V13	0.9818	0.9994	0.9725	0.9903	0.9832	0.9878
V14	0.9600	1.0036	0.9541*	0.9977	0.9508*	0.9663

settings are enforced over the whole planning period. It can be seen that even when applying the greedy-optimum control settings, reactive production at node 2 (Qg2) violates its limits. In Table III, these results are also compared to the results of the Q-learning (RL) [13] and the probabilistic CLF method [9], obtained for the same network. The Q-learning algorithm [13] provides slightly worse results, since Qg2 violates its limits, and the voltage at node 14 violates its lower limit, too. The absolute value of maximum total reward (mtr) [13] in this case was calculated at 0.804, which is greater than the corresponding index of the ACS algorithm $(\mathrm{mtf} = 0.732)$. The Q-learning algorithm [13] took about 38 800 iterations to find the greedy-optimum control settings. In the case of probabilistic CLF [9], the upper limit of reactive production at node 2 (Qg2) and the lower limit of voltage at node 14 as well as the upper limit of the apparent flow (T23) in line 2-3 are violated.

One way of enforcing violated limits of Qg2 is to relax the constant voltage limit at node 2, considering it as a PQ bus and allowing the voltages at nodes 6 and 1 to be set at 1.021 and 1.03 p.u., respectively [9]. Rerunning the ACS algorithm under these new considerations, the best solutions are provided as shown in Figs. 4(a)–(c), in the nominal, heavy, and light loads, respectively. Among them, the greedy-optimum control settings that achieve the mtf (10) over the whole planning period are once more the optimal control settings obtained for the nominal load [see Fig. 4(a)]. In this case, convergence of the ACS algorithm took 2645 iterations. Table IV shows the mtf (10) calculated at 0.557, the greedy optimum control settings, and the operating limits of constrained variables when these settings are enforced over the whole planning period. In Table IV, the greedy-optimum control settings are also compared to the results of the Q-learning [13] and the probabilistic CLF method [9]. In the case of Q-learning algorithm [13], the corresponding absolute value (mtr) was calculated at 0.565, which is almost equal to mtf, $(\mathrm{mtf} = 0.557)$. It must be underscored that the Q-learning algorithm took about 42 800 iterations to find the greedy-optimum control settings [13].

The ACS and Q-learning algorithms provide the optimal results, rather than the near-optimal results given by the probabilistic CLF method [9], since all constraints, including the upper limit of apparent flow (T23) on line 2-3, are satisfied.

Table V shows the optimal settings proposed by ACS algorithm at five operating points (11) corresponding to the average load values $(\mathrm{k} = 0)$ and the two adjacent pairs $(\mathrm{k} = \pm 1, \pm 2)$ together with the maximum total rewards of the optimal actions.

A key advantage of the proposed ACS algorithm is its flexibility in providing control actions that can satisfy additional criteria and, thus, solve multicriteria optimization problems. For example, if the cost of VAr compensation should be taken into account, then as greedy-optimal action, the action that minimizes compensation at node 9 could be chosen.

Table VI shows the operating space of constrained variables when the new optimal settings are enforced over the whole planning period. The convergence of the ACS algorithm to optimal actions in the case of minimum VAr compensation corresponding to operating points $\mathrm{k} = -2$ and $\mathrm{k} = -1$ (see Table V) takes 2563 and 2021 iterations, respectively. These results show that the ACS algorithm provides control settings for the whole planning period and can be more effective than

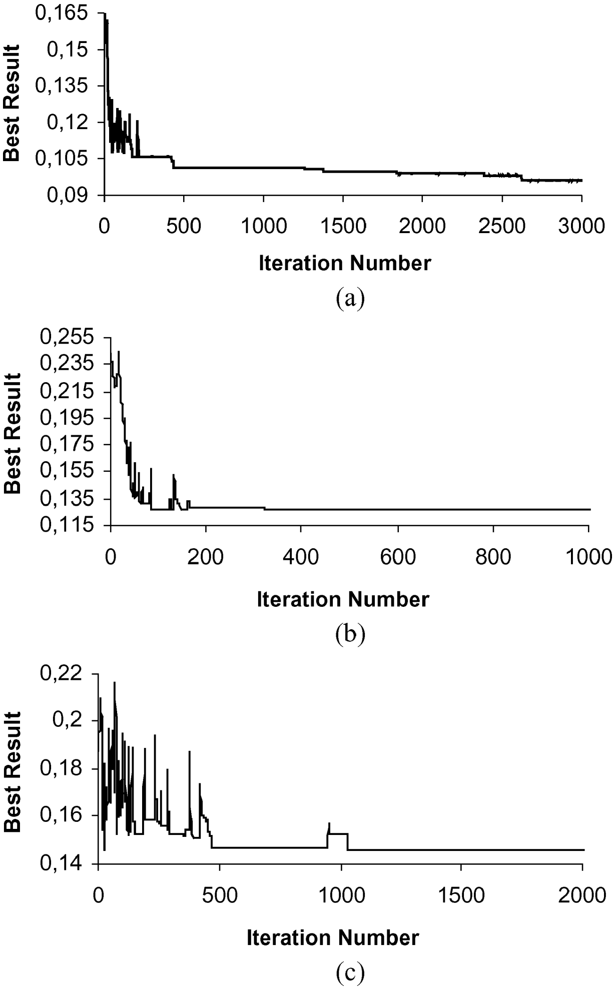

Fig. 4. Performance of ACS algorithm when Qg2 is Cutoff in the (a) nominal, (b) heavy, and (c) light loads.

TABLE IV
COMPARISON OF RESULTS BETWEEN ACS, RL, AND PROBABILISTIC LOAD FLOW ON THE IEEE 14-BUS SYSTEM

	ACS Algorithm		Q-Learning Algorithm		Probabilistic Load Flow	
	mtf = 0.557		*mtr* = -0.565		-	
Contr Varib	Optimal Settings		Greedy-Optimal Settings (a*)		Optimal Settings	
*t*56	0.99		0.99		0.94	
*t*49	0.90		0.91		0.97	
*t*47	0.99		0.97		0.98	
*b*9	0.06		0.03		0.12	
*V*6	1.021		1.021		1.021	
*V*1	1.030		1.030		1.030	
Const Varib	Wmin	Wmax	Wmin	Wmax	Wmin	Wmax
Qg3	0.0132	0.5475	0.0100	0.5595	0.5120	0.5695
Qg6	0.0434	0.3770	0.0462	0.3794	0.3066	0.4214
T23	0.2182	0.6758	0.2180	0.6767	0.7063	0.7665*
T56	0.0332	0.2577	0.0330	0.2569	0.4126	0.4936
V4	0.9788	1.0060	0.9772	1.0044	0.9726	0.9822
V5	0.9898	1.0128	0.9888	1.0118	0.9763	0.9843
V7	0.9822	1.0241	0.9891	1.0311	0.9797	0.9954
V8	0.9822	1.0241	0.9891	1.0311	0.9797	0.9954
V9	0.9801	1.0284	0.9807	1.0289	0.9749	0.9933
V10	0.9780	1.0230	0.9786	1.0235	0.9737	0.9903
V11	0.9938	1.0191	0.9940	1.0192	0.9926	1.0013
V12	0.9997	1.0112	0.9998	1.0112	1.0008	1.0038
V13	0.9904	1.0086	0.9905	1.0086	0.9932	0.9979
V14	0.9600	1.0058	0.9603	1.0062	0.9605	0.9765

TABLE V
OPTIMAL ACTIONS OVER THE WHOLE PLANNING PERIOD (CUTOFF QG2)

k	Operating points	Optimal settings				
		*t*56	*t*49	*t*47	*b*9	*mtf*
- 2	*μ – 6σ/20*	1.01	0.94	0.93	0.00	0.565
- 1	*μ – 3σ/20*	0.99	0.92	0.93	0.00	0.558
0	*μ*	0.99	0.90	0.99	0.06	0.557
1	*μ + 3σ/20*	0.99	0.90	0.99	0.06	0.559
2	*μ + 6σ/20*	0.99	0.90	1.01	0.06	0.563

TABLE VI
OPERATING SPACE OF CONSTRAINED VARIABLES WHEN THE CRITERION IS MINIMUM VAR COMPENSATION AT BUS 9

Actions	Optimal settings			
	k = -2		k = -1	
*t*56	1.01		0.99	
*t*49	0.94		0.92	
*t*47	0.93		0.93	
*b*9	0.00		0.00	
*V*6	1.021		1.021	
*V*1	1.03		1.03	
Constrained Variables	**Wmin**	**Wmax**	**Wmin**	**Wmax**
Qg3	0.0179	0.6201	0.0967	0.6377
Qg6	0.0628	0.4530	0.0871	0.3778
T23	0.2278	0.6858	0.2289	0.6878
T56	0.1113	0.2875	0.0556	0.2580
V4	0.9739	0.9944	0.9709	0.9914
V5	0.9876	1.0039	0.9837	1.0000
V7	1.0052	1.0421	1.0062	1.0431
V8	1.0052	1.0421	1.0062	1.0431
V9	0.9845	1.0280	0.9876	1.0312
V10	0.9817	1.0228	0.9843	1.0254
V11	0.9957	1.0189	0.9970	1.0202
V12	1.0000	1.0111	1.0003	1.014
V13	0.9911	1.0085	0.9915	1.0090
V14	0.9627	1.0056	0.9647	1.0076

the probabilistic CLF method [9] since it satisfies constraints with minimum VAr compensation.

The ACS algorithm is also applied to the reactive power control problem for the IEEE 136-bus system. This system consists of 136 buses (33 PV and 103 load buses), 199 lines, 24 transformers, and 17 reactive compensations. In our study, the ACS algorithm learns the optimum control settings for each of 41 operating points selected from the whole planning period similarly to the IEEE 14-bus case [9]. The control variables selected comprise voltages at PV buses 4 and 21 (discrete variations 0.99 to 1.02, in step 0.01), taps at transformers 28, 41, and 176 (discrete variation of 0.92 to 1.00, in steps of 0.02), and reactive compensation (b) at buses 3 and 52 (discrete variation of six blocks). The total number of actions is $5^2 \times 5^3 \times 6^2 = 112\,500$. The constrained variables include voltages at all PQ buses (from 0.96 to 1.05 p.u.) and three apparent power flows at the most heavily loaded lines 156 and 177 (upper limit 4.6 p.u.) and 179 (upper limit 3.4 p.u.). The initial control settings violate the power flow limits of all the above lines and upper limit of the voltages of buses 18, 19, and 23. The ACS algorithm learns the greedy-optimal control action, resulting in the satisfaction of the limits of constrained variables over the whole planning period, as shown in Table VII.

In this case, the agent found the optimum control action at the average load values after about 2910 iterations (see Fig. 5) in contrast to 112 980 iterations of Q-learning algorithm. The

TABLE VII
RESULTS OF ACS AND Q-LEARNING ALGORITHMS ON THE IEEE 136-BUS SYSTEM

Control Actions	**ACS algorithm**		**Q-Learning algorithm**	
	mtf = 0.628		*mtr* = -0.640	
	Optimal Settings		Greedy-Optimal settings (a*) (p.u.)	
*V*4	1.01		1.00	
*V*21	0.99		0.99	
*t*28	0.94		0.92	
*t*41	0.92		0.92	
*t*176	0.92		0.92	
*b*3	0.17		0.15	
*b*52	0.17		0.17	
Constrained Variables	**Wmin**	**Wmax**	**Wmin**	**Wmax**
V18	0.990	1.027	0.987	1.021
V19	0.998	1.029	0.998	1.028
V23	1.012	1.050	1.010	1.049
T156	3.986	4.500	3.987	4.500
T177	3.948	4.501	3.948	4.501
T179	2.561	3.223	2.675	3.234

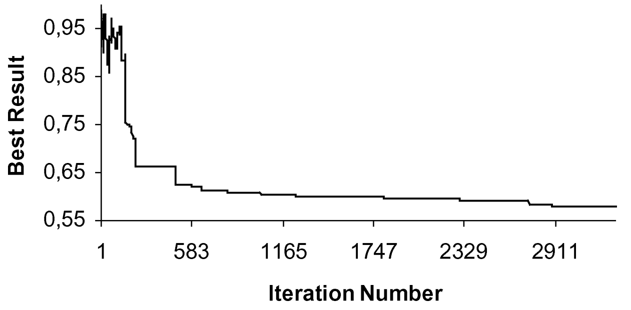

Fig. 5. Performance of ACS algorithm in the average values of loads of IEEE 136-bus system.

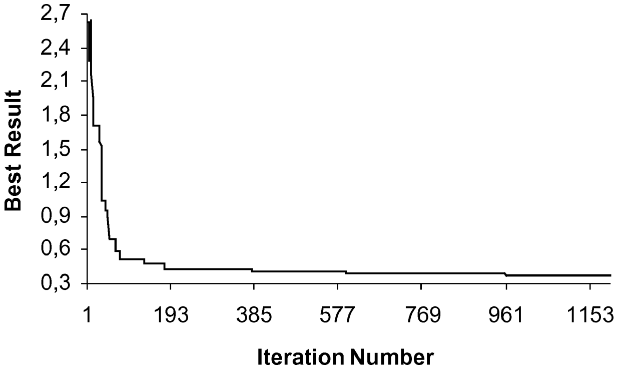

Fig. 6. Greedy-optimum performance of the ACS algorithm in reactive power optimization of the IEEE 14-bus system.

TABLE VIII
RESULTS OF ACS ON THE IEEE 14-BUS SYSTEM WITH POWER LOSSES AS OBJECTIVE FUNCTION

ACS		
Power losses over the whole planning period = [0.11467, 0.01513] pu		
Control Variable	**Optimal Settings**	
*t*56	0.98	
*t*49	0.91	
*t*47	1.01	
*b*9	0.09	
*V*6	1.02	
*V*1	1.03	
Constraint Variable	**Wmin**	**Wmax**
Qg3	0.1531	0.5938
Qg6	0.1483	0.4394
T23	0.2268	0.6835
T56	0.1009	0.2837
V4	0.9784	0.9989
V5	0.9903	1.0067
V7	0.9887	1.0255
V8	0.9887	1.0255
V9	0.9850	1.0289
V10	0.9822	1.0235
V11	0.9959	1.0192
V12	1.0001	1.0112
V13	0.9911	1.0086
V14	0.9631	1.0062

total computing time is about 8 s on a 1.4-GHz Pentium-IV PC, compared to the 160 s achieved by Q-learning [13].

Summarizing, ACS and the Q-learning algorithms provide the optimal results, rather than the near-optimal results provided by the probabilistic CLF method [9]. A key advantage of the proposed ACS algorithm is its flexibility in providing control actions that can accommodate additional criteria and, thus, solve multicriteria optimization problems. The main advantage of ACS algorithm in comparison with the Q-learning algorithm [13] is the better results in greatly less number of iterations.

As another application, the ACS algorithm (see Table I) can be applied to the reactive power control problem for the IEEE 14-bus system by considering as objective function both the real power losses and the operating constraints expressed by (5). The AS-graph updates its pheromone by selecting a large number of load/generation combinations (1000 operating points) over the whole planning period. However, these 1000 uncorrelated operating points are sampled randomly with normal and discrete distribution probabilities of the loads and generations [9]. The control variables comprise all transformer taps (t), the reactive compensation (b) at bus 9, and the generator voltages of buses 1 and 6.

Among all operating points, the best takes 961 iterations to find the optimum control action, as shown in Fig. 6. The optimal settings and the voltages over the whole planning period are shown in Table VIII. The real power losses calculated over the whole planning period are between 0.015 13 and 0.114 67 p.u., compared to the initial losses which were between 0.0843 and 0.2067 p.u. and those given by the probabilistic CLF, which were between 0.0242 and 0.1387 p.u. Consequently, the results are better than the initial and the ones obtained by the probabilistic CLF (see Table IV) in minimizing the real power losses while satisfying all operating constraints.

The total number of load flows is equal to (the number of iterations reported) × (the number of ants) for every one of the randomly selected operating points. In the case of the greedy-optimum operating point of the IEEE 14-bus system, load flow is run $961 \times 100 = 96\,100$ times. Fig. 6, like the rest of the figures in the paper, shows the best solution out of 100 ants at each iteration. However, in terms of computing time for 100 ants, the 961 iterations are still too much. The number of iterations can be further reduced by determining the optimum parameters (M, Q, R, a) in the ACS algorithm. This can be achieved by incorporating any of the modern evolutionary algorithms such as cultural algorithms [28] in the proposed ACS.

VI. Conclusion

In this paper, the ACS approach is applied to the solution of the CLF problem. The CLF problem is modified as a combinatorial optimization problem. An iterative ACS algorithm is implemented, providing the optimal offline control settings over a planning period, satisfying all operating limits of the constrained variables. Our algorithm consists of mapping the solution space, expressed by an objective function of the combinatorial optimization CLF problem on the space of control settings, where artificial ants walk. Test results show that the ACS algorithm can be used to find optimum solutions within a reasonable time. The results of the proposed algorithm are also compared to those obtained by the Q-learning and probabilistic CLF methods. The approach is very flexible, allowing its application to multicriteria optimization problems, e.g., assigning priorities to control actions. Moreover, the ACS algorithm was implemented in the reactive power control problem by considering the real power losses and operating constraints over the whole planning period. These results demonstrate the superiority of the ACS algorithm in providing the optimal control settings. In order to increase the speed of convergence, an evolutionary algorithm, such as the cultural-ACS algorithm, will be introduced in the future to determine the optimum values of the empirical parameters of the ACS algorithm.

References

[1] W. Kellermann, H. M. Zein El-Din, C. E. Graham, and G. A. Maria, "Optimization of fixed tap transformer settings in bulk electricity systems," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1126–1132, Aug. 1991.

[2] J. G. Vlachogiannis, "Control adjustments in fast decoupled load flow," *Elect. Power Syst. Res.*, vol. 31, no. 3, pp. 185–194, Dec. 1994.

[3] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A practical swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. Power Syst.*, vol. 15, no. 4, pp. 1232–1239, Nov. 2000.

[4] P. K. Satpathy, D. Das, and P. B. Dutta Gupta, "A novel fuzzy index for steady state voltage stability analysis and identification of critical busbars," *Elect. Power Syst. Res.*, vol. 63, no. 2, pp. 127–140, Sep. 2002.

[5] B. Cova, N. Losignore, P. Marannino, and M. Montagna, "Contingency constrained optimal reactive power flow procedures for voltage control in planning and operation," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 602–608, May 1995.

[6] E. Vaahedi, Y. Mansour, J. Tamby, W. Li, and D. Sun, "Large scale voltage stability constrained optimal VAr planning and voltage stability applications using existing OPF/Optimal VAr planning tools," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 65–74, Feb. 1999.

[7] D. Gan, R. J. Thomas, and R. D. Zimmerman, "Stability constrained optimal power flow," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 535–540, May 2000.

[8] R. A. Jabr and A. H. Coonick, "Homogeneous interior point method for constrained power scheduling," *Proc. Inst. Elect. Eng., Gener., Transm., Distrib.*, vol. 147, no. 4, pp. 239–244, Jul. 2000.

[9] T. S. Karakatsanis and N. D. Hatziargyriou, "Probabilistic constrained load flow based on sensitivity analysis," *IEEE Trans. Power Syst.*, vol. 9, no. 4, pp. 1853–1860, Nov. 1994.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, Aug. 1992.

[11] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, Jan.–Jun. 1996.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction, Adaptive Computations and Machine Learning*. Cambridge, MA: MIT Press, 1998.

[13] J. G. Vlachogiannis and N. D. Hatziargyriou, "Reinforcement learning for reactive power control," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1317–1325, Aug. 2004.

[14] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico de Milano, Milan, Italy, 1992.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[16] M. Dorigo and L. M. Cambardella, "Ant colony system: a cooperative learning approach to traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[17] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristics," in *Proc. Congr. Evol. Comput.*, vol. 2, 1999, pp. 1470–1477.

[18] P. Merz and B. Freisleben, "A comparison of memetic algorithm, tabu search and ant colonies for the quadratic assignment problem," in *Proc. Congr. Evol. Comput.*, vol. 3, 1999, pp. 1999–2070.

[19] I.-K. Yu, C. S. Chou, and Y. H. Song, "Application of the ant colony search algorithm to short-term generation scheduling problem of thermal units," in *Proc. Int. Conf. Power Syst. Technol.*, vol. 1, 1998, pp. 552–556.

[20] S.-J. Huang, "Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches," *IEEE Trans. Energy Conver.*, vol. 16, no. 3, pp. 296–301, Sep. 2001.

[21] J.-H. Teng and Y.-H. Liu, "A novel ACS-based optimum switch relocation method," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 113–120, Feb. 2003.

[22] Y.-J. Jeon, J.-C. Kim, S.-Y. Yun, and K. Y. Lee, "Application of ant colony algorithm for network reconfiguration in distribution systems," in *Proc. IFAC Symp. Power Plants Power Syst. Control*, Seoul, Korea, Jun. 9–12, 2003, pp. 266–271.

[23] T. Stützle and H. H. Hoos, "Improvements on the ant system: introducing the MAX-MIN ant system," in *Artificial Neural Networks and Genetic Algorithms*, R. F. Albrecht, G. D. Smith, and N. C. Steele, Eds. New York: Springer Verlag, 1998, pp. 245–249.

[24] R. N. Allan and M. R. G. Al-Shakarchi, "Probabilistic techniques in a.c. load-flow analysis," *Proc. Inst. Elect. Eng., Gener., Transm., Distrib.*, vol. 124, no. 2, pp. 154–160, Feb. 1977.

[25] J. F. Kenney and E. S. Keeping, *Mathematics in Statistics*. Princeton, NJ: Nostrand, 1962.

[26] D. Ernst and L. Wehenkel, "FACTS devices controlled by means of reinforcement learning algorithms," in *Proc. PSCC*, Sevilla, Spain, Jun. 22–24, 2002.

[27] D. Ernst, M. Glavic, and L. Wehenkel, "Power systems stability control: reinforcement learning framework," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 427–435, Feb. 2004.

[28] B. Franklin and M. Bergerman, "Cultural algorithms: concepts and experiments," in *Proc. Congr. Evol. Comput.*, 2000, pp. 1245–1251.

John G. Vlachogiannis received the Diploma in electrical engineering and Ph.D. degree from Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1990 and 1994, respectively.

He founded the Laboratories for Industrial and Energy Informatics (IEI-Lab) and Total Quality Management (TQM-Lab), R.S. Lianokladiou, Lamia, Greece. Also, he is currently an Assistant Professor at Informatics and Computer Technology Department, Technological Educational Institute of Lamia. His research interests include control and management strategies and artificial intelligence techniques applied in the planning and operation of power and industrial systems.

Dr. Vlachogiannis is a member of the Greek Computer Society (Member of IFIP, CEPIS), a member of the Hellenic Artificial Intelligence Society (EETN), a member of the Technical Chamber of Greece, and member of World Engineers (WEN).

Nikos D. Hatziargyriou (S'80–M'82–SM'90) was born in Athens, Greece, in 1954. He received the Diploma in electrical and mechanical engineering from National Technical University of Athens (NTUA), Athens, Greece, in 1976 and the M.Sc and Ph.D degrees from University of Manchester Institute of Science and Technology, Manchester, U.K., in 1979 and 1982, respectively.

He is currently a Professor with the Power Division of the Electrical and Computer Engineering Department, NTUA. His research interests include dispersed generation and renewable energy sources, dynamic security, and artificial intelligence techniques.

Dr. Hatziargyriou is a member of CIGRE SCC6 and member of the Technical Chamber of Greece.

Kwang Y. Lee (F'01) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1964, the M.S. degree in electrical engineering from North Dakota State University, Fargo, in 1968, and the Ph.D. degree in system science from Michigan State University, East Lansing, in 1971.

He has been with Michigan State, Oregon State, University of Houston, and the Pennsylvania State University, where he is now a Professor of Electrical Engineering and Director of Power Systems Control Laboratory. His interests include power system control, operation, planning, and intelligent system applications to power systems and power plant control.

Dr. Lee is an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS and an Editor of IEEE TRANSACTIONS ON ENERGY CONVERSION. He is also a registered Professional Engineer.