Optimization of Power Systems based on Ant Colony System Algorithms: An Overview

Kwang Y. Lee, *Fellow IEEE* and John G. Vlachogiannis

***Abstract* -- This study presents the Ant Colony System (ACS) algorithms for optimization of power systems planning. The developed ACS algorithms formulate complex problems as combinatorial optimization problems. They are distributed algorithms composed by a set of cooperating artificial agents, called ants, which cooperate to find an optimum solution of the combinatorial problems. 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 combinatorial problems on the space of control variables (Ant System (AS)-graph), where ants walk. In this study an ACS algorithm is applied to the constrained load flow (CLF) problem on IEEE 14-bus test system and 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, ACS algorithm is applied to the reactive power control problem on the IEEE 14-bus test system in order to minimize the real power losses subject to operating constraints over the whole planning period. Finally, the application of ACS algorithm for active/reactive operational planning of power systems on IEEE 30-bus test system is presented and results are compared to those given by Simulated Annealing (SA), exhibiting superior performance.**

***Index Terms* -- Ant Colony System (ACS), combinatorial optimization, constrained load flow, reactive power control, active/reactive operational planning.**

I. INTRODUCTION

THERE are large number of different combinatorial optimization problems facing electricity utilities. The deregulation of electricity supply industry world-wide adds ever growing motivations to develop new optimization algorithms. Therefore best strategies are designed for most effectiveness utilizing the asset under increasing commercial pressure [1]-[2]. Various algorithmic and heuristic approaches [3]-[4] have been adopted or investigated by power engineers. These include the lambda-iteration method, gradient method, Lagrangian relaxation, benders decomposition, interior point method, linear/nonlinear programming and dynamic programming, etc. More recently heuristic techniques such as artificial neural networks, simulated annealing, tabu-search, reinforcement learning and evolutionary computing have also been intensively investigated. In particular, for the last fifteen years there has been a growing interest in algorithms inspired by the observation of the natural process and behavior of natural creatures, such as the Inheritance of Cultural among Human Generations, Quantum Mechanics Computing, Particle Swarm Optimization (PSO) and Ant Colony Systems (ACS) to help solve complex problems.

Among them the last has been proofed that it handles successfully various combinatorial complex problems. Dorigo has proposed the first ACS in his Ph.D. thesis [5]. The ACS method belongs to biologically inspired heuristics (meta-heuristics) methods. Real ants are capable of finding the shortest path from 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 pheromone previously deposited by other ants. For one ant, the path is chosen according to the quantity of pheromone. Furthermore, this chemical substance has a decreasing action over time, and the quantity left by one ant depends on the amount food found and the number of ants using this trail [6]. This behavior has inspired the ACS algorithm in which a set of artificial ants cooperate in solving a problem by exchanging information via pheromone deposited on a graph.

Ant Colony Systems (ACS) algorithms have recently been introduced as powerful tools to solve the order based problems such as quadratic assignment problem [6],[7] and traveling salesman problem (TSP) [8]-[10] and other real-world problems, such as the web usage mining [11], optimal reconfiguration of distribution systems [12], optimal placement of capacitors in distribution systems [13], efficient combinational circuit synthesis [14]. In power systems the ACS has been applied to solve scheduling problems including the unit commitment and economic dispatch problems [15]-[24], optimum switch relocation and network reconfiguration problems for distribution systems [25]-[27] and planning problems [28]-[30]. For most of these applications, the results show that the ACS-based approach can outperform other heuristic methods.

This study focuses on the applications of ACS algorithms to the combinatorial optimization problems faced by the power systems such as constrained load flow (CLF), reactive power control and active/reactive planning. Specifically:

a) The problem of the off-line control settings has been

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

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

tackled by the probabilistic CLF formulation [31]. The method [30] takes into account load uncertainties and generating unit unavailability modeled as probability density functions. However, this method provides near-optimum off-line control settings. Recent research solves the CLF problem by means of heuristic reinforcement learning (RL) method [32]-[34]. The RL modified the CLF problem as a combinatorial optimization problem [35]. In this study, 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 [35]. In this study, the CLF problem is solved by means of the ACS algorithm [28]. 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 this 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 [35] reward function. The objective function "fires" the transition function, which gives the probability for an ant to select an edge to walk. In this study, for computational simplicity, the transition function considers only the trail intensity for the transition probability [5],[25],[35], i.e., more ants choose has more probability to be selected [28]. The results are compared with those obtained by the probabilistic CLF [31] and the Q-learning method [35], showing the superiority of the proposed ACS algorithm [28].

b) The reactive power control is formulated as a combinatorial optimization problem and calculate using ACS algorithm [28]. The most common methods, are: Classical methods (such as NLP), have drawbacks in that they don't satisfactorily handle nonconvexities and nonsmoothness such as generator's prohibited operating zones, operating constraints of the transmission lines such as thermal limits and switchable VAR source constraints. On the contrary, ACS algorithm [28] handles satisfactorily previously mentioned operating problems and does not require that the objective functions and the constraints have to be differentiable and continuous. The results are obtained by ACS [28] on the IEEE 14-bus test system.

c) Finally, the active/reactive operational planning is solved by means of the ACS algorithm [29]. Specifically in this study, ACS algorithm [29] aims to determine the optimal settings of voltage control variables, such as generator outputs, voltages, transformer taps and shunt VAR compensation devices, considered as nodes of Ant-System (AS) graph. Results are compared to those given by meta-heuristic technique of Simulated Annealing (SA) [36] for the network of IEEE 30-bus test system, exhibiting superior performance.

The study is organized in six sections. Section II describes the basic concepts of a general ACS algorithm for combinatorial optimization problems. Section III formulates the CLF, reactive power control problems and active/reactive operational planning while the proposed ACS algorithm for handling them is presented in Section IV. In Section V the proposed ACS algorithm is applied to CLF and reactive power control problems on IEEE 14-bus test system and 136 bus system as well as in active/reactive operational planning on IEEE 30-bus test system. Finally, in Section VI, general conclusions are drawn.

II. ACS Algorithms for Combinatorial Optimization Problems

ACS algorithms are developed based on the observation of foraging behavior of real ants. Although they are almost blind animals with very simple individual capacities, they can find the shortest route between their nest(s) and a source of food. An ant group is based on the principle that using simple communication mechanisms 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 towards 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. In the ACS algorithms, a set of artificial ants walk on a construction Ant System graph (AS-graph) following certain rules and cooperate in solving a problem by exchanging information via pheromone trails deposited on the edges of the AS-graph. The crucial part of the ACS algorithms is that each artificial ant (ant hereafter) works individually but exchanges information implicitly with other ants via the pheromone trails which can be accessed and altered by all eligible ants. All ants can take into consideration their own and other ants' experiences when they decide their future movement so that they act together in a colony to achieve an optimal solution.

ACS algorithms are suitable for solving combinatorial optimization problems $\zeta = (S, f, \Omega)$ which can be defined as [1]:

A set of discrete variables X_i with values $x_i \in D_i = \{d^i_1, d^i_2, \ldots, d^i_{|Di|}\}$, $i = 1,2,\ldots,n$, where d^i_j, $1 \le j \le |D_i|$ is a candidate solution component for variable X_i, the subscript $|D_i|$ indicates the number of candidate components for the variable X_i and n is the number of discrete variables involved in the problem; A set Ω of constraints among the variables; An objective function $f : D_1 \times D_2 \ldots \times D_n \to \mathrm{R}$, where R represents the set of real numbers, to be minimized. The set of all feasible solutions is [1]:

$$S = \{s = \{(X_1, x_1), (X_2, x_2), \ldots, (X_n x_n)\} \mid x_i \in D_i, i \in [1, n], s.t.\ \Omega\} \tag{1}$$

Each element in S can be seen as a candidate solution and S is called the search space. In order to solve the combinatorial optimization problem $\zeta = (S, f, \Omega)$, a corresponding construction graph, in which the space that the artificial ants will walk, should be mapped on according to the problem under consideration. It has been experienced that finding an appropriate graph representation for the problem to be solved is one of the most difficult (and ad hoc) tasks to face when applying the ACS algorithms to solve real-world problems [1].

The pheromone trails for all connections (edges) in the construction AS-graph are initialized to be a pre-determined value. Each ant in the colony randomly sets its initial solution, s_i, $j=1,2,\ \ldots,\ M$, within the given feasible region before the ants start to construct their solutions. Based on their fitness, ants deposit pheromone on the edges composing their solution paths. The structure of a general ACS algorithm is shown in the following major steps [1] (Table I):

TABLE I
GENERAL ACS ALGORITHM

1. Initialize pheromone trails and place *M* ants on the nodes of AS graph
2. Repeat until system convergence
2.1 For *i* = 1 to *n*
2.1.1 For *j* = 1 to *M*
2.1.1.1 Choose the node *s* to move to, according to the transition probability (2)
2.1.1.2 Move the ant-*k* to the node *s*
2.2 Update the pheromone using the pheromone update formula (3)

Step 1: Initialize *A(t):* The decision variables to be optimized, X_i, i = 1,2,…n, are identified and their eligible values are determined to form the feasible region (construction graph). A colony of M ants is generated. Each ant gives an initial solution s_j, j= 1,2,…M within the feasible region. The maximum iteration number in each run is M and the maximum construction step number in each iteration is n. The pheromone trails for all the connections in the graph are initialized to be a pre-determined value. The termination criteria of each run and each iteration are respectively set up. The objective/fitness function is defined in terms of the decision variables.

Step 2: Evaluate *A(t):* The fitness of all ants are evaluated based on their initial solutions.

Step 3: Alter the pheromone trails: Each ant adds pheromone, consisting of its solution in proportion to its fitness to the path.

Step 4: Send ants *A(t):* According to the fitness function, ants' performance will be weighed in terms of their fitness values which influences the quantity of pheromone deposited on the trails that they have passed. At the beginning of each iteration, all ants are located on their initial positions on the graph. Each ant chooses the next node to move by taking into account two parameters: the desirability of the nodes and the pheromone trails previously deposited by other ants. In this operation, each ant selects its next move using tournament selection based on the above two parameters. The k-th ant allocated at the node r decides to move to the node s on the basis of probability $p_k(r,s)$, which is shown as follows:

$$p_k(r,s)=\begin{cases}\dfrac{\left[\gamma(r,s)\right]^{\alpha}\cdot\left[\eta(r,s)\right]^{\beta}}{\sum_{u\in N_r^k}\left[\gamma(r,u)\right]^{\alpha}\cdot\left[\eta(r,u)\right]^{\beta}} & if\quad s\in N_r^k \\ 0 & otherwise\end{cases} \tag{2}$$

where $\gamma(r,s)$ represents the pheromone trail associated to the connection between nodes r and s, $\eta(r,s)$ a heuristic value, called the desirability of adding connection $l_{r,s}$ or node r to the k-th ant's partial solution and can be determined according to the optimization problem under consideration. $\eta(r,s)=\left|\mu\text{-}\Delta \mathrm{F}\right|$ and ΔF = original total cost – new total cost. N_r^k is the feasible neighbor components of the k-th ant at the node r with respect to the problem constraints Ω, α ($0<\alpha<1$) is the pheromone decay parameter and β is a parameter which determines the relative importance of pheromone versus the distance ($\beta > 0$).

Step 5: Pheromone updating: Once all ants complete their solutions after n steps, the pheromone trails are updated by:

$$\gamma(r,s)=(1-\alpha)\cdot\gamma(r,s)+\sum_{k=1}^{m}\Delta\gamma_k(r,s) \tag{3}$$

where: $\Delta\gamma_k(r,s)$ is the quantity of pheromone deposited on the edge between the nodes r and s during the iteration, i.e., n moves.

Thereafter, many other ACS algorithms based on previously described general version have been published for handling combinatorial optimization problems. The basic of them are briefly described [37]:

- *ANTS:* According to [38] the term ANTS derives from the fact that the proposed algorithm can be interpreted as an Approximate Nondeterministic Tree Search since it shares several elements with an approximated branch and bound procedure. The most significant modification is the use of lower bounds on the solution cost of the completion of a partial solution to compute dynamically changing heuristic values η_{ij}. Further modifications are the use of a different action choice rule and a modified trail update rule [38].
- *MMAS:* Max-Min Ant System [37] is an improvement over general ACS presenting three following modification: First, to exploit the best solutions found in an iteration, after each iteration only one ant (global best or iteration best ant) is allowed to add pheromone. Second, to avoid search stagnation, the allowed range of the pheromone trail strengths is limited. Third, the pheromone trails are initialized to the upper trail limit, which causes a higher exploitation at the start of the algorithm. In a following section we present these modifications in more detail.
- *Fast Ant system (FANT):* FANT [39] is an improved version of ANTS. In FANT solutions are constructed in

the same way as in MMAS. FANT differs from MMAS in the number of ants used and the management of pheromone trails.

- *Ant-Q:* Ant-Q algorithm [40] exploits connections of ACS algorithm with Q-learning [35]. Ant-Q differs in three main aspects from general ACS. First, Ant-Q uses a more aggressive action choice rule than general ACS. Second, the pheromone is added only to trails belonging to the global-best solution. Third, each time an ant uses a trail to move from point to point it removes some pheromone from this trail.
- *HAS:* Although Hybrid Ant System algorithm [41] is inspired from FANT, it has a major difference that the pheromone trails are not used to construct new solutions but to modify the current solutions.

III. Combinatorial Optimization Problems of Power Systems

The formulation of power systems' problems handling as combinatorial optimizations by ACS algorithms are following:

A. Constrained Load Flow

The load flow problem can be expressed by the next two sets of non-linear equations:

$$\begin{aligned} Y &= g(X,U) \\ Z &= h(X,U) \end{aligned} \tag{4}$$

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. In Section V of this paper a technique to maintain constrained variables within operating limits over the whole planning period using the ACS algorithm is proposed.

B. Reactive Power Control

The reactive power control is to optimize the steady state performance of a power system in terms of one or more objective functions while satisfying several equality and inequality constraints. The problem can be generally formulated as follows. The objective is to minimize the real power losses in transmission lines that can be expressed as:

$$P_{Loss}(X,\ U) = \sum_{l=1}^{Nl} P_l \tag{5}$$

where P_l is the real power losses at line-*l*, Nl is the number of transmission lines, and X and U are the state and control vectors, respectively.

Specifically, the vector of state or dependent variables, X, consists of load bus voltages V_L, generator reactive power outputs Q_G, and transmission line loadings S_l. Hence, X can be expressed as:

$$X^T = \left[V_{L_1}, V_{L_2} ... V_{L_{Nd}}, Q_{G_1}, Q_{G_2} ... Q_{G_{Ng}}, S_{L_1}, S_{L_2} ... S_{L_{NL}}\right] \tag{6}$$

The vector of control variables, U, consists of generator voltages V_G, transformer tap settings T, and shunt VAR compensations Q_C. Hence, U can be expressed as:

$$U^T = \left[V_{G_1}, V_{G_2} ... V_{G_{Ng}}, Q_{C_1}, Q_{C_2} ... Q_{C_{NC}}, T_1, T_2 ... T_{NT}\right] \tag{7}$$

The constraints represent the system operating constraints as follows:

Generation constraints: Generator voltages V_G and reactive power outputs Q_G are restricted by their limits as follows:

$$V_{G_i}^{\min} \le V_{G_i} \le V_{G_i}^{\max} \quad i = 1,2,...,NG \tag{8}$$

$$Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max} \quad i = 1,2,...,NG \tag{9}$$

where NG is the number of generators.

Switchable VAR source constraints: Switchable VAR compensations Q_C are restricted by their limits as follows:

$$Q_{C_i}^{\min} \le Q_{C_i} \le Q_{C_i}^{\max} \quad i = 1,2,...,NC \tag{10}$$

where NC is the number of switchable VAR sources.

Transformer constraints: Transformer tap settings T are bounded as follows:

$$T_i^{\min} \le T_i \le T_i^{\max} \quad i = 1,2...,NT \tag{11}$$

where NT is the number of transformers.

Security constraints: These include the constraints of load voltages at load buses V_L and transmission line loadings S_l as follows:

$$V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max} \quad i = 1,2...,Nd \tag{12}$$

$$S_{l_i} \le S_{l_i}^{\max} \quad i = 1,2...,Nl \tag{13}$$

where Nd and Nl are the number of nodes and lines of power system, respectively.

C. Active/Reactive Operational Planning

Modern power systems have many operations such as the dispatch of active power and others known as ancillary services. Active/reactive operational planning belongs to this category of services. It allocates Volt control and reactive support in accordance with open market mechanisms [36]. The problem of active/reactive operational planning is formulated as an optimization problem with objective function expressed by the following equation for comparison reasons with Simulated Annealing (SA) technique [36]:

$$f = \sum_{i=1}^{Ng} C_i^t(P_{gi}^t) + \sum_{d=1}^{Nd}\left(V_d^t - V_d^{\max}\right)\cdot p(V) + \sum_{d=1}^{Nd}\left(V_d^{\min} - V_d^t\right)\cdot p(V) + \sum_{j=1}^{Nl}\left(\frac{I_j^t}{I_j^{\max}} - 1\right)\cdot p(I) \tag{14}$$

subject to the mild constraints expressed by (4), (8)-(13)

Here, C_i^t is the active power cost of unit-*i* at time-*t,* P_{gi}^t is active power generation of unit-*i* at time-*t, Ng* is the total number of units, *Nd* is the total number of buses, *Nl* is the set of network branches, $V_d^{\min}$, $V_d^{\max}$ are the limits of voltage at bus-*d*, and $I_j^{\max}$ is the thermal limit of the transmission line-*j*.

The penalty factors *p(V)* and *p(I)* enforce the voltage and thermal limits:

$$p(V) = \begin{cases} 40 & \text{if } V_d^t > V_d^{\max} \ or \ V_d^t < V_d^{\min} \\ 0 & \text{else} \end{cases} \tag{15}$$

$$p(I) = \begin{cases} 300 & \text{if } I_j^t > I_j^{\max} \\ 0 & \text{else} \end{cases} \tag{16}$$

IV. MMAS Algorithm for the Combinatorial Optimization Problems of Power Systems

Among state-of-the-art ACS algorithms the most popular one which handles successfully plethora of combinatorial optimization problems is the Max-Min AS (MMAS) algorithm [37]. In this study we improve the MMAS version in order to handle the combinatorial optimization problems of power systems. The ants simulate the transitions from one point *r* to another point *s*, according to the proposed MMAS algorithm [37] 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 [37]:

$$p(r,s) = \frac{\gamma(r,s)}{\sum_l \gamma(r,\ell)} \qquad s, \ell \in N_r^k \tag{17}$$

where matrix *γ*(*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:

$$\gamma(r,s) = \alpha\cdot\gamma(r,s) + \Delta\gamma^k(\mathrm{r,s}) \tag{18}$$

where *α* with 0<*α*<1, is the persistence of the pheromone trail, so that (1-*α*) 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 (18). The randomness implies that pheromone is deposited in all possible trails, not just in the best trail. However, the trail with favorable update 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 following paragraphs the MMAS algorithm is extended and modified to solve the CLF and reactive power control problems as well as the active/reactive operational planning, farther called simply combinatorial optimization problems.

The settings of control variables (tap-settings, VAr compensation blocks, *etc.*) are combined in order to achieve the power system constraints. In this approach, the graph that describes the settings of control variables of the combinatorial optimization problems 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 combinatorial optimization problems. All possible candidate discrete settings for a control variable are represented by the states *r* of the AS-graph (*r = 1, 2,…, m*). The control variables are represented by the stages *i* (*i = 1, 2,…*, *n*), where *n* is the number of the control variables. Each ant starts its tour at the home colony and stops at the destination. The proposed 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 AS graph is created. 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, ACS-algorithm achieves the best exploration of 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 (17). When the ant *k* moves from one stage to the next, the state of each stage will 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 (18) (local update). For the purpose of this research, the pheromone update $\Delta\gamma^k(r,s)$ is chosen as:

$$\Delta\gamma^{k}(r,s) = \frac{1}{Q \cdot f} \tag{19}$$

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

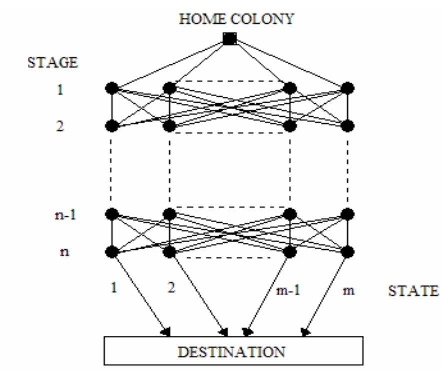


Fig. 1. Search space for the combinatorial optimization problems.

Specially, 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:

$$f = \frac{1}{n} \cdot \sum_{j=1}^{n} \left| \frac{2z_j - z_{j\max} - z_{j\min}}{z_{j\max} - z_{j\min}} \right| \tag{20}$$

where n expresses the number of constrained variables, z_j the value of *j-th* 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 in CLF problem [35]. The objective function f has this formulation in order 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 is calculated for each run. Then, the pheromone trails (r,s) of the best ant tour (ant with minimum objective function) is updated (global update) as:

$$\gamma(r,s) = \alpha \cdot \gamma(r,s) + \frac{R}{f_{best}} \qquad r,\ s\ \in J_{best}^{k} \tag{21}$$

where R is a large positive constant. Both Q in (21) and R are arbitrarily large numbers. Empirical tests have shown that the ACS-algorithm converges faster when Q is almost equal to R.

TABLE II
ACS ALGORITHM IN COMBINATORIAL OPTIMIZATION PROBLEMS OF POWER SYSTEMS

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 (23), 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 number of iterations 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 (17).
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 appropriate objective function ((20) in case of CLF, (14) in case of active/reactive operational planning, (5) in case of reactive power control) for each ant.
4.1.5 Update the pheromone of (r,s)-trails for each ant, using the local pheromone update formulae (18), (19).
4.1.6 Update the pheromone of (r,s)-trails belonging to best ant tour (f_{best}), using the pheromone update formula (21).
4.1.7 In order to avoid the ants stagnations, enforce the limits (22)-(24).
4.2 In the case of CLF and reactive power control problems enforce each of the best control settings over the whole planning period and calculate (25).
5. In the case of CLF and reactive power control problems choose as a greedy-optimum control setting the one that minimizes (25).

b) To avoid search stagnation (the situation where all the ants follow the same path, that is, they construct the same solution [37]), the allowed range of the pheromone trail strengths is limited to:

$$\gamma\ (r,s) = \begin{cases} \tau_{\min} & \text{if } \gamma\ (r,s) \le \tau_{\min} \\ \tau_{\max} & \text{if } \gamma\ (r,s) \ge \tau_{\max} \end{cases} \tag{22}$$

For our study the limits are chosen as:

$$\tau_{\max} = \frac{1}{\alpha \ \cdot f_{gbest}} \tag{23}$$

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

$$\tau_{\min} = \frac{\tau_{\max}}{M^2} \tag{24}$$

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 multi-objective function (*mtf*) over the whole planning period is defined as a greedy-optimum control setting:

$$mtf = \min\{total \text{ of objective functions}\} = \min\left(\sum_{\text{over whole planning period}} f\right) \quad (25)$$

Table II shows the execution steps of the MMAS algorithm applied to the CLF, reactive power control problems and active/reactive operational planning.

V. RESULTS

A. ACS applied to CLF problem [28]

The proposed ACS algorithm is applied to adjust reactive control variables in the IEEE 14-bus test system shown in Fig. 2. The system consists of the slack bus (node 1), three PV (nodes 2, 3 and 6), ten PQ buses and twenty branches. It has been used in many probabilistic studies. The network data and load probabilistic data are the same as used in [31]. 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 3 to 5 impulses each and 8 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, 4 banks at node 2, 2 banks at node 3 and 2 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 (Fig. 2).

The upper part of Table III shows the limits of the control variables *Umin* and *Umax* and the discrete steps in variation. The transformer taps (*t*56, *t*49, and *t*47) are in 16 steps, while the reactive compensation (*b*9) is in 9 steps. Therefore, the last step of *b*9 (*b*9 = 0.24) is repeated for the next 7 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 III the upper and lower limits of all constrained variables *Wmax* and *Wmin* are shown.

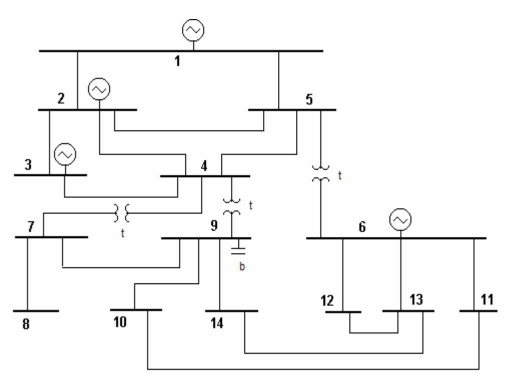


Fig. 2. Line diagram of the IEEE 14-bus system.

In this study, the following parameters are chosen: M = 100, n = 4, m = 16, 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 α = 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 to 3000 iterations. The ACS algorithm can be implemented in a large number of load combinations (operating points) selected over the whole planning period.

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

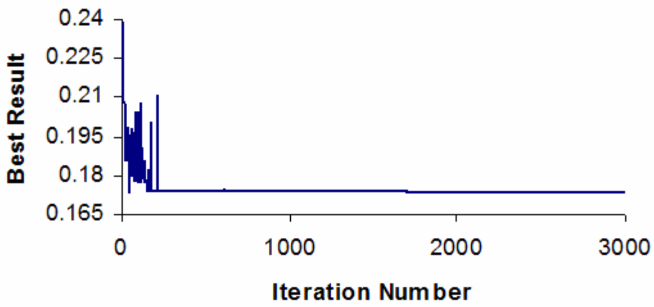
Control Actions	Umin	Umax	Step
*t*56	0.90	1.05	0.01
*t*49	0.90	1.05	0.01
*t*47	0.90	1.05	0.01
*b*9	0.00	0.24	0.03
Constrained Variables	Wmin	Wmax	
Qg2	0.00	0.30	
Qg3	0.00	0.70	
Qg6	0.00	0.45	
T23	0.00	0.75	
T56	0.00	0.50	
V4	0.96	1.05	
V5	0.96	1.05	
V7	0.96	1.05	
V8	0.96	1.05	
V9	0.96	1.05	
V10	0.96	1.05	
V11	0.96	1.05	
V12	0.96	1.05	
V13	0.96	1.05	
V14	0.96	1.05	

In this 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:

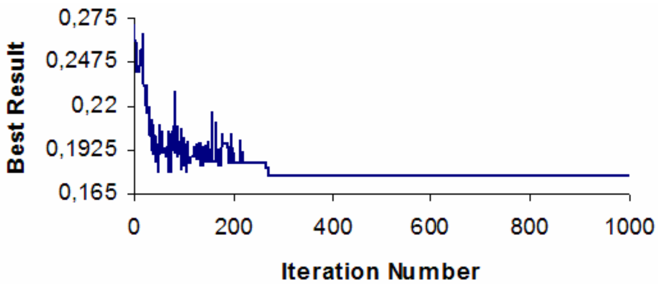
$$\text{Load step} = \left(\mu \pm k \times \frac{3\sigma}{20}\right) \cdot 100\% \quad (26)$$

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 [42]. The μ and σ for the discrete distributions of loads are calculated using the formulae given by [43].

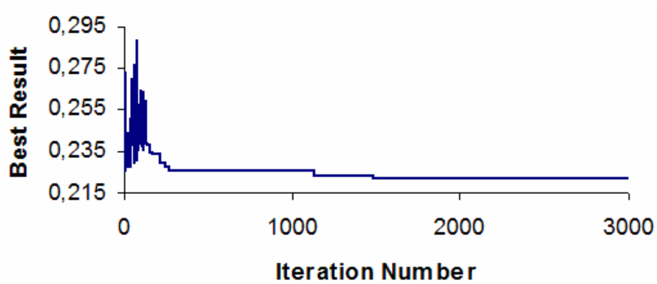
Performance of the ACS algorithm is shown in Figs. 3a, 3b, and 3c depicting the obtained values of objective function (20) during the ACS procedure for the nominal, heavy and light load, respectively. The nominal load corresponds to the mean values of the load. The heavy and light load correspond to the 1% confidence limit that all load values are lower and higher than these values, respectively. The Figs. 3a-3c show the convergence of ACS algorithm in a minimum value of (20), achieving the optimum control settings for each of the 3 operating points corresponding to the heavy, light and nominal load.



(a)



(b)



(c)

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

Among the 41 optimal control settings, the *greedy-optimum* control settings are those that provide the minimum total function (25). Tests have shown that the calculation of (25) in the above 3 operating points is sufficient to provide the minimum total function (*mtf*). Results on IEEE 14-bus show that the greedy-optimum control settings that achieve the *mtf* (10) over the whole planning period is the optimal control settings obtained for the nominal load (Fig 3a). In this case, convergence of the ACS algorithm took 1,730 iterations. Table IV shows the *mtf* (25) is calculated at 0.732, the greedy-optimum control settings and the operating space of constrained variables, when these 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 the Table IV these results are also compared to the results of the Q-learning (RL) [35] and the probabilistic CLF method [31], obtained for the same network. The Q-learning algorithm [35] 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*) [35] in this case was calculated at 0.804, which is greater than the corresponding index of the ACS algorithm (*mtf* = 0.732). The Q-learning algorithm [35] took about 38,800 iterations to find the greedy-optimum control settings.

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 = 0.732		*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

In the case of probabilistic CLF [31] 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 p.u. and 1.03 p.u., respectively [31]. Rerunning the ACS algorithm under these new considerations the best solutions are provided as shown in Figs. 4a, 4b and 4c, in the nominal, heavy and light loads, respectively. Among them the greedy-optimum control settings that achieve the *mtf* (25) over the whole planning period are once more the optimal control settings obtained for the nominal load (Fig. 4a). In this case, convergence of the ACS algorithm took 2,645 iterations. Table V shows the *mtf* (25) 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 V the greedy-optimum control settings are also compared to the results of the Q-learning [35] and the probabilistic CLF method [31]. In the case of Q-learning algorithm [35], the corresponding absolute value (*mtr*) was calculated at 0.565, which is almost equal to *mtf*, (*mtf* = 0.557). It must be underscored that the Q-learning algorithm took about 42,800 iterations to find the greedy-optimum control settings [35].

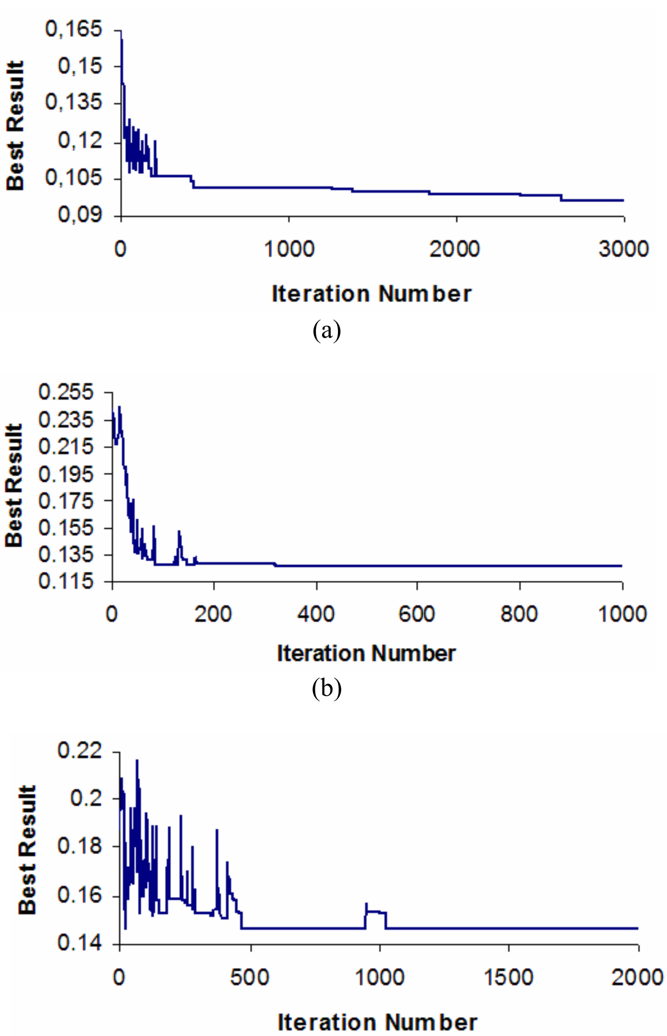


Fig. 4. Performance of ACS algorithm when Qg2 is Cutoff in the (a) nominal, (b) heavy and (c) light loads.

The ACS and Q-learning algorithms provide the optimal results, rather than the near-optimal results given by the probabilistic CLF method [31], since all constraints including the upper limit of apparent flow (T23) on line 2-3 are satisfied.

Table VI shows the optimal settings proposed by ACS algorithm at 5 operating points (26) corresponding to the average load values (k=0) and the two adjacent pairs (k=±1, ±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 multi-criteria 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 V
COMPARISON OF RESULTS BETWEEN ACS, RL AND PROBABILISTIC LOAD FLOW ON THE IEEE 14-BUS SYSTEM

	ACS Algorithm *mtf* = 0.557		**Q-Learning Algorithm** *mtr* = -0.565		**Probabilistic Load Flow** -	
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 VI
OPTIMAL ACTIONS OVER THE WHOLE PLANNING PERIOD (CUTOFF QG2)

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

Table VII shows the operating space of constrained variables when the new optimal settings are enforced over the whole planning period. The convergence of ACS algorithm to optimal actions in the case of minimum VAr compensation corresponding to operating points k = -2 and k = -1 (Table VI), 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 the probabilistic CLF method [31] since it satisfies constraints with minimum VAr compensation.

TABLE VII
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	1.0140
V13	0.9911	1.0085	0.9915	1.0090
V14	0.9627	1.0056	0.9647	1.0076

The ACS algorithm is also applied to the CLF problem for the 136-bus system [44],[45]. This system consists of 136 buses (33 PV and 103 load buses), 199 lines, 24 transformers and 17 reactive compensations. In this 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 [31]. 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 6 blocks). The total number of actions is $5^2x5^3x6^2$= 112,500. The constrained variables include voltages at all PQ buses (from 0.96 p.u. to 1.05 p.u.) and 3 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 VIII. In this case, the agent found the optimum control action at the average load values after about 2910 iterations (Fig. 5) in contrast to 112,980 iterations of Q-learning algorithm. The total computing time is about 8 sec on a 1.4 GHz Pentium-IV PC, compared to the 160 sec achieved by Q-learning [35].

Summarizing, ACS and the Q-learning algorithms provide the optimal results, rather than the near-optimal results provided by the probabilistic CLF method [31]. A key advantage of the proposed ACS algorithm is its flexibility in providing control actions that can accommodate additional criteria and thus solve multi-criteria optimization problems. The main advantage of ACS algorithm in comparison with the Q-learning algorithm [35] is the better results in greatly less number of iterations.

TABLE VIII
RESULTS OF ACS AND Q-LEARNING ALGORITHMS ON THE 136-BUS SYSTEM REACTIVE POWER CONTROL

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

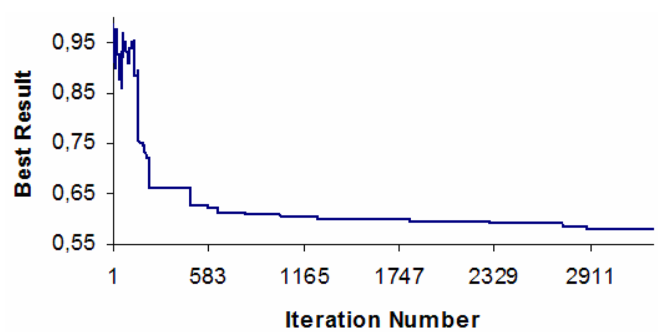


Fig. 5. Performance of ACS algorithm in the average values of loads of 136-bus system.

B. ACS applied to Reactive Power Control [28]

As another application, ACS algorithm (Table II) 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 (5) and operating constraints expressed by (8)-(13) as these are included in (20). The AS-graph updates its pheromone by selecting 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 [31]. The control variables comprise all transformer taps (*t*), the reactive compensation (*b*) at bus 9, and the generator voltages of buses 1 and 6.

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

ACS algorithm Power losses over the whole planning period = [0.11467, 0.01513] pu		
Control Variables	**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 Variables	**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

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 IX. The real power losses calculated over the whole planning period are between 0.01513 p.u. and 0.11467 p.u., compared to the initial losses which were between 0.0843 p.u. and 0.2067 p.u. and those given by the probabilistic CLF which were between 0.0242 p.u. and 0.1387 p.u. Consequently, the results are better than the initial and the ones obtained by the probabilistic CLF (Table V) 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) x (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 x 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 [46] in the proposed ACS.

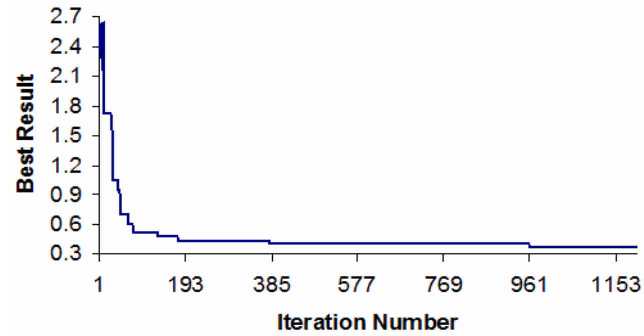


Fig. 6. Greedy-optimum performance of the ACS algorithm in reactive power optimization of the IEEE-14 bus system.

C. ACS applied to Active/Reactive Operational Planning [29]

The proposed ACS algorithm is applied on the active/reactive operational planning of IEEE 30-bus test system. The line diagram of this network is given in Fig. 7. Some modifications in network's data of IEEE 30-bus test system are made in order to be the same with those given in [36] for comparison purposes. Specifically, the network consists of 4 generators, 41 lines, 4 transformers and 2 capacitor banks. In the transformer tests, 7 tap positions in each transformer were considered. Each position corresponds to 0.02 increments within the interval [0.94, 1.06]. The available reactive powers of capacitor banks are [0, 7.5, 15, 22.5, 30] MVAr and they are connected to buses 10 and 24. Generator voltages are discretized in 150 steps (step: 0.0006 pu) within the range of [0.96, 1.05]. Loads were set at the values referred in [47], multiplied by a factor of 0.6 (nominal load). The increment/decrement accuracy for the generator outputs was set to 1MW/0.01 pu.

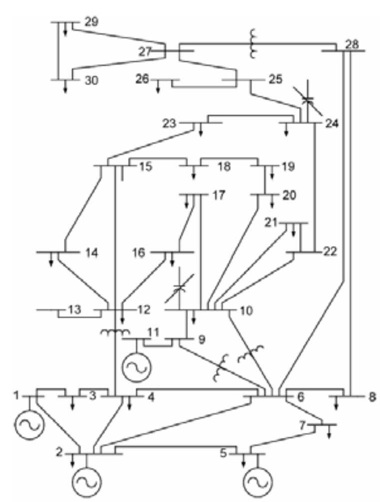


Fig. 7. Line diagram of the IEEE 30-bus system [36].

The bid curves of four generators and the minimum and maximum submitted capacities are given by [36]:

$$\text{bid}(P_1) = \begin{cases} 10€/\text{MWh} & 50\text{MW} \le P_1 < 80\text{MW} \\ 20€/\text{MWh} & 80\text{MW} \le P_1 < 110\text{MW} \\ 30€/\text{MWh} & 110\text{MW} \le P_1 < 140\text{MW} \\ 40€/\text{MWh} & 140\text{MW} \le P_1 < 170\text{MW} \\ 50€/\text{MWh} & 170\text{MW} \le P_1 \le 200\text{MW} \end{cases} \quad (27)$$

$$\text{bid}(P_2) = \begin{cases} 10€/\text{MWh} & 20\text{MW} \le P_2 < 36\text{MW} \\ 20€/\text{MWh} & 36\text{MW} \le P_2 < 52\text{MW} \\ 30€/\text{MWh} & 52\text{MW} \le P_2 < 68\text{MW} \\ 40€/\text{MWh} & 68\text{MW} \le P_2 < 84\text{MW} \\ 50€/\text{MWh} & 84\text{MW} \le P_2 < 100\text{MW} \end{cases} \quad (28)$$

$$\text{bid}(P_3) = \begin{cases} 10€/\text{MWh} & 10\text{MW} \le P_3 < 18\text{MW} \\ 20€/\text{MWh} & 18\text{MW} \le P_3 < 26\text{MW} \\ 30€/\text{MWh} & 26\text{MW} \le P_3 < 34\text{MW} \\ 40€/\text{MWh} & 34\text{MW} \le P_3 < 42\text{MW} \\ 50€/\text{MWh} & 42\text{MW} \le P_3 \le 50\text{MW} \end{cases} \quad (29)$$

$$\text{bid}(P_4) = \begin{cases} 10€/\text{MWh} & 3\text{MW} \le P_4 < 5.4\text{MW} \\ 20€/\text{MWh} & 5.4\text{MW} \le P_4 < 7.8\text{MW} \\ 30€/\text{MWh} & 7.8\text{MW} \le P_4 < 10.2\text{MW} \\ 40€/\text{MWh} & 10.2\text{MW} \le P_4 < 12.6\text{MW} \\ 50€/\text{MWh} & 12.6\text{MW} \le P_4 \le 15\text{MW} \end{cases} \quad (30)$$

In this study, the following ACS parameters are chosen: $M = 300$, $n = 14$, $m = 150$, $Q = R = 5{,}000{,}000$ and the initial best solution is estimated at 0.1. The parameter α in (3) from our experience shows that any value in the range [0.88, 0.999] works well. In this paper it is chosen as $\alpha = 0.99$. In this study, the search will terminate 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 to 3000 iterations.

The ACS algorithm converges in 2010 iterations (Fig. 8) and the final value of objective function (14) is 3050€. The final value given by Simulated Annealing (SA) method [36] is 3141€. An improvement of 91€ is obtained compared to the evaluation given by [36].

The final settings of control variables for ACS and SA are given in Table X. It is shown that the generator outputs given by ACS algorithm are slightly different from those given by SA [36].

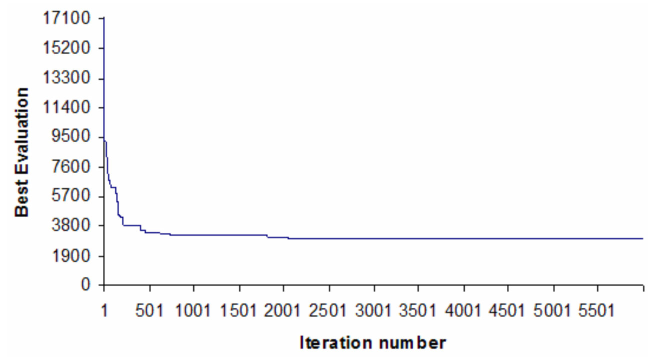


Fig. 8. Performance of ACS algorithm in the nominal load of IEEE 30-bus test system.

TABLE X
ACS AND SA [36] SETTINGS OF CONTROL VARIABLES FOR IEEE 30-BUS TEST SYSTEM.

Control Variables	ACS algorithm	SA algorithm
P_{G1} (MW)	109	108
P_{G2} (MW)	35	35
P_{G5} (MW)	25	25
P_{G11} (MW)	5	6
V_{G1}	1.037	1.000
V_{G2}	1.022	1.020
V_{G5}	1.012	1.010
V_{G11}	1.041	1.020
T_{6-9}	1.00	1.02
T_{6-10}	0.94	0.98
T_{4-12}	0.94	1.04
T_{27-28}	1.02	0.98
QC_1 (MVAr)	0.0	0.0
QC_2 (MVAr)	15.0	7.5

The bus voltages and apparent flows in pu are given in Figs. 9 and 10, respectively. The bus voltages are within the acceptable voltage range of [0.96, 1.05] as shown in Fig 9. According to Fig. 10 all branch apparent flows are much lower than the acceptable ranges of 202 MW / 2.02 pu (for 132 KV lines between buses: 1-2, 1-3, 2-4, 2-5, 2-6, 3-4, 4-6, 6-28, 5-7, 6-8, 6-7, 8-28) and 30 MW / 0.3 pu (for 33 KV lines).

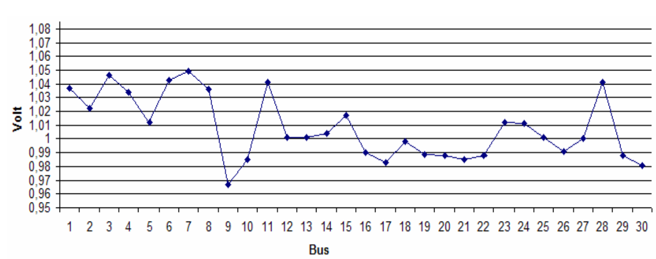


Fig. 9. Bus voltages.

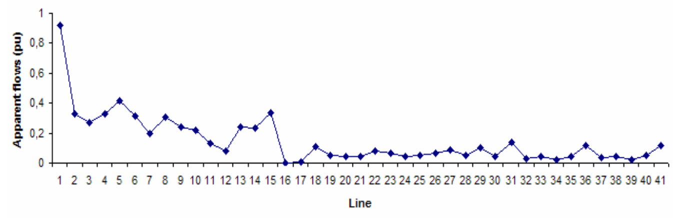


Fig. 10. Apparent flows (pu).

VI. CONCLUSIONS

In this study the optimization of power systems planning based on state-of-the-art Ant Colony System (ACS) algorithms are reviewed. The most efficient and flexible of them (*etc.*, MMAS) is applied to the solution of the constrained load flow (CLF) and reactive power control problems as well as to active/reactive operational planning of power systems. These are modified as combinatorial optimization problems. An iterative ACS algorithm is implemented, providing the optimal off-line control settings over a planning period satisfying all operating limits of the constrained variables. The algorithm consists of mapping the solution space, expressed by three objective functions of the combinatorial optimization problems on the space of control settings, where artificial ants walk. Test results show that the proposed ACS algorithm can be used to find optimum solutions within a reasonable time. The results of the proposed ACS algorithm are also compared to those obtained by Q-learning, probabilistic CLF and the Simulated Annealing (SA) methods. The approach is very flexible allowing its application to multi-criteria optimization problems, *e.g.*, assigning priorities to control actions. In order to increase the speed of convergence an evolutionary algorithm, such as Cultural-ACS algorithm, will be introduced in future to determine the optimum values of empirical parameters of the ACS algorithm.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Professor Anastasios G. Bakirtzis at Aristotle University of Thessaloniki, Greece for providing the model of the 136-bus system.

REFERENCES

[1] Y.H. Song, K.Y. Lee, J.G. Vlachogiannis, Y.H. Lu and I.K. Yu "Ant colony search algorithms in power system optimization," *In "Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems,"* Eds. K.Y. Lee and M.A. El-Sharkawi, *IEEE Press Book,* (to appear).

[2] Y.H. Song (Ed.), *Modern Optimisation Techniques in Power Systems,* Kluwer Academic Publishers, 1999.

[3] M.R. Irving, Y.H. Song, "Optimisation techniques for electrical power systems," *IEE Power Engineering Journal,* Vol. 14, No. 5, 2000, pp. 245-254.

[4] Y.H. Song, M.R. Irving, "Optimisation techniques for electrical power systems Part 2 Heuristic optimisation techniques," *IEE Power Engineering Journal,* Vol. 15, No. 3, 2001, pp. 151-160.

[5] M. Dorigo, *Optimization, learning and natural algorithms,* PhD Thesis, Politecnico de Milano, Italy, 1992.

[6] E.-G. Talbi, O. Roux, C. Fonlupt and D. Robillard, "Parallel ant colonies for the quadratic assignment problem," *Future Generation Computer Systems,* Vol. 17, 2001, pp. 441-449.

[7] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimisation by a colony of co-operating agents," *IEEE Trans. on Systems, Man and Cybernetics Part B,* 1996, Vol. 26, Issue 1, pp. 29-41.

[8] P. Wei, W.Q. Xiong and J.Y. Zhao, "An improved ant colony algorithm for TSP," *In Fifth World Congress on Intelligent Control and Automation,* 2004 (WCICA 2004), Vol. 3, 15-19 June 2004, pp. 2263-2267.

[9] J. Ouyang and G.-R. Yan, "A multi-group ant colony system algorithm for TSP," *In Proc 2004 International Conference on Machine Learning and Cybernetics,* Vol. 1, 26-29 Aug. 2004, pp. 117-121.

[10] M. Dorigo and L.M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. on Evolutionary Computation,* Vol. 1, Issue 1, April 1997, pp. 53-66

[11] A. Abraham and V. Ramos, "Web usage mining using artificial ant colony clustering and linear genetic programming," *The 2003 Congress on Evolutionary Computation,* (CEC '03), Vol. 2, 8-12 Dec. 2003, pp. 1384-1391.

[12] E. Carpaneto and G. Chicco, "Ant-colony search-based minimum losses reconfiguration of distribution systems," *In Proc of the 12th IEEE Mediterranean Electrotechnical Conference,* 2004 (MELECON 2004), Vol. 3, 12-15 May 2004, pp. 971-974.

[13] R. Annaluru, S. Das and A. Pahwa, Multi-level ant colony algorithm for optimal placement of capacitors in distribution systems, *In Congress on Evolutionary Computation,* 2004 (CEC2004), Vol. 2, 19-23 June 2004, pp. 1932-1937.

[14] B.A.B. Sarif, M. Abd-El-Barr, S.M. Sait and U. Al-Saiari, "Fuzzified ant colony optimization algorithm for efficient combinational circuits synthesis," *In Congress on Evolutionary Computation,* 2004 (CEC2004), Vol. 2, 19-23 June 2004, pp. 1317-1324.

[15] Libao Shi, Jin Hao, Jiaqi Zhou and Guoyu Xu, "Short-term generation scheduling with reliability constraint using ant colony optimization algorithm," *In Fifth World Congress on Intelligent Control and Automation,* 2004, (WCICA 2004), Vol. 6, 15-19 June 2004, pp. 5102-5106.

[16] S. Kopuri and N. Mansouri, "Enhancing scheduling solutions through ant colony optimization," *In Proc of the 2004 International Symposium on Circuits and Systems,* 2004 (ISCAS '04), Vol. 5, 23-26 May 2004, pp. V-257 - V-260.

[17] T. Sum-im, and W. Ongsakul, "Ant colony search algorithm for unit commitment," *In IEEE Intrn Conf on Industrial Technology,* Vol. 1, 10-12 Dec. 2003, pp. 72-77.

[18] Shyh-Jier Huang, "Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches," *IEEE Trans on Energy Conversion,* Vol. 16, Issue 3, Sept. 2001, pp. 296-301.

[19] Yun-He Hou, Yao-Wu Wu, Li-Juan Lu and Xin-Yin Xiong, "generalized ant colony optimization for economic dispatch of power systems," *In Proc of Intern Conf on Power System Technology,* 2002, (PowerCon 2002), Vol. 1, 13-17 Oct. 2002, pp. 225-229.

[20] N.S. Sisworahardjo and A.A.El-Keib, "unit commitment using the ant colony search algorithm," *In Large Engineering Systems Conf on Power Engineering,* 2002 (LESCOPE 02), 26-28 June 2002, pp. 2-6.

[21] In-Keun 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 of 1998 Intern Conf on Power System Technology* (POWERCON '98), Vol. 1, 18-21 Aug. 1998, pp. 552-556.

[22] Y.H. Song, C.S. Chou and Y. Min, "Large-scale economic dispatch by artificial ant colony search algorithms," *Electric Machines and Power Systems,* Vol. 27, No. 5, 1999, pp. 679-690.

[23] I.K. Yu and Y.H. Song, "A novel short-term generation scheduling technique of thermal units using ant colony search algorithms," *Electric Power and Energy Systems,* Vol. 23, 2001, pp. 471-479.

[24] Y.H. Song, C.S. Chou and T.J. Stonham, "Combined heat and power economic dispatch by improved ant colony search algorithm," *Electric Power Systems Research,* Vol. 52, 1999, pp. 115-121.

[25] J.-H. Teng and Y.-H. Liu, "A novel ACS-based optimum switch relocation method", *IEEE Trans. on Power Systems,* Vol. 18, No. 1, Feb. 2003, pp. 113-120.

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

[27] M.G. Ippolito, E.R. Sanseverino and F. Vuinovich, "Multiobjective ant colony search algorithm optimal electrical distribution system planning," *In Congress on Evolutionary Computation,* 2004 (CEC2004), Vol. 2, 19-23 June 2004, pp. 1924-1931.

[28] J.G. Vlachogiannis, N.D. Hatziargyriou and K.Y. Lee, "Ant colony system based algorithm for constrained load flow problem," *IEEE Trans. on Power Systems,* Vol. 20, No. 3, Aug. 2005, pp. 1241-1249.

[29] K.Y. Lee and J.G. Vlachogiannis, "Ant colony optimization for active/reactive operational planning," *In 16th World IFAC Conf,* July 4-8, 2005, Prague, Czech, paper code: 6.3-02415.

[30] Y.H. Song and C.S. Chou, "Application of ant colony search algorithms in power system optimisation," *IEEE Power Engineering Review,* Vol. 18, No. 12, 1998, pp. 63-64.

[31] T. S. Karakatsanis and N. D. Hatziargyriou, "Probabilistic constrained load flow based on sensitivity analysis", *IEEE Trans. Power Systems,* Vol. 9, No. 4, November 1994, pp. 1853-1860.

[32] C.J.C.H. Watkins and P. Dayan, "Q-learning," *Machine learning,* Vol. 8, No. 3, Aug. 1992, pp. 279-292.

[33] L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research,* Vol. 4, Jan.-June 1996, pp. 237-285.

[34] R.S. Sutton and A.G. Barto, "Reinforcement learning: An introduction, adaptive computations and machine learning," Cambridge, MA: MIT Press, 1998.

[35] J.G. Vlachogiannis and N.D. Hatziargyriou, Reinforcement learning for reactive power control, *IEEE Trans. on Power Systems,* Vol. 19, No. 3, Aug. 2004, pp. 1317-1325.

[36] K.A. Papadogiannis, N.D. Hatziargyriou and J.T. Saraiva, "Short term active/reactive operation planning in market environment using simulated annealing," *In Proc. ISAP03,* Aug. 31 - Sept. 3, 2003, Lemnos, Greece.

[37] T. Stützle and H. H. Hoos, "MAX-MIN ant system and local search for combinatorial optimization problems," *In* S. Voss, S. Martello, I.H. Osman and C. Roucairol (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization,* Kluwer Academic, Boston 1999, pp. 313-329.

[38] V. Maniezo, "Exact and approximate nondeterministic tree-search procedure for the quadratic problem," Technical report CSR-98-1, C.L. in Scienze dell' Informazione, Universita di Bologna, Sole di Cesena, Italy, 1998.

[39] E.D. Taillard and L.M. Cambardella, "Adaptive memories for the quadratic assignment problem," Technical report IDSIA-87-97, Lugano, Switzerland, 1997.

[40] L.M. Gambardella and M. Dorigo, "Ant-Q: A Reinforcement Learning approach to the traveling salesman problem," *In Proceedings of the Eleventh Intern Conf. on Machine Learning*, pp. 252-260, Morgan Kaufmann, 1995.

[41] L.M. Gambardella, E.D. Taillard and M. Dorigo, "Ant colonies for QAP," Technical report IDSIA-4-97, Lugano, Switzerland, 1997. Accepted for publication in the *Journal of the Operational Research*.

[42] R. N. Allan and M. R. G. Al-Shakarchi, "Probabilistic techniques in a.c. load-flow analysis", *IEE Proc. Gener., Transm., Distrib.,* Vol. 124, No. 2, February 1977, pp. 154-160.

[43] J.F. Kenney and E.S. Keeping, "Mathematics in Statistics," Princenton, NJ, Nostrand, 1962.

[44] J.G. Vlachogiannis, "Control adjustments in load flow studies," *Electric Power Systems Research,* Vol. 31, No. 3, pp. 185-194, Sept. 1994.

[45] J.G. Vlachogiannis, "Contribution to load flow analysis and control of power systems," *Ph.D. Dissertation,* Aristotle University of Thessaloniki, Greece, July 1994.

[46] B. Franklin and M. Bergerman, "Cultural algorithms: Concepts and experiments", *In Proc. 2000 Congr. Evol. Computation,* pp. 1245-1251.

[47] The IEEE 30-bus test system. Available online in: *http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.htm*.

Kwang Y. Lee received his B.S. degree in Electrical Engineering from Seoul National University, Korea, in 1964, M.S. degree in Electrical Engineering from North Dakota State University, Fargo, in 1968, and Ph.D. degree in System Science from Michigan State University, East Lansing, in 1971. He has been with Michigan State, Oregon State, Univ. 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 a Fellow of IEEE, Associate Editor of IEEE Transactions on Neural Networks, and Editor of IEEE Transactions on Energy Conversion. He is also a registered Professional Engineer.

John G. Vlachogiannis received his Diploma in Electrical Engineering and Ph.D. degree both from Aristotle University of Thessaloniki, Greece, in 1990 and 1994, respectively. He is a Head of the Laboratories for Industrial and Energy Informatics (*IEI-Lab*) and Total Quality Management *(TQM-Lab)* at *R.S. Lianokladiou, 35100, Lamia, Greece.* His research interests include control and management strategies and artificial intelligence techniques applied in planning and operation of power and industrial systems. Dr. Vlachogiannis is a member of the Greek Computer Society (Member of *IFIP, CEPIS*), member of the Hellenic Artificial Intelligence Society *(EETN)*, member of the Technical Chamber of Greece and member of World Engineers *(WEN)*.