Application of Key Cutting Algorithms for Optimal Power Flow Problems

Uthen Leeton[1] and **Thanatchai Kulworawanichpong**[2], Non-members

ABSTRACT

This paper illustrates an application of Key Cutting Algorithm (KCA) to optimal power flow (OPF) problems in comparative with some effective mathematical and evolutionary optimization methods. The KCA is one of intelligence algorithms (AI), which was just developed since 2009. This algorithm emulates the work of locksmiths to open the lock. The best key that matches a given lock is pretended to be an optimal solution of a relevant optimization problem. The basic structure of the key cutting algorithm is as simple as that of genetic algorithms in which a string of binary numbers is employed as a key to open the lock. The proposed algorithm was tested with four mathematical test functions and three standard IEEE test power systems (6-bus, 14-bus and 30-bus test systems). The test power systems were divided into two cases. The first test case was given by applying a quadratic function to generators' fuel-cost curve whereas a non-smooth fuel-cost function was assigned to the second. The comparisons among solutions obtained by sequential quadratic programming (SQP), genetic algorithms (GA), particle swarm optimization (PSO) and key cutting algorithm (KCA) were carried out. As revealed from the simulated results, the effectiveness of the KCA algorithm for solving OPF problems was confirmed.

Keywords: Optimal Power Flow, Genetic Algorithm, Particle Swarm Optimization, Sequential Quadratic Programming, Key Cutting Algorithm

1. INTRODUCTION

To date, an electrical power system is very large and obviously complicated due to technological enhancement of power system engineering. Increasing of electrical energy consumption often leads extending and upgrading an existing power transmission and distribution network to serve all customers sufficiently, effectively and economically. To achieve good performance for power delivery, a real power dispatching problem must be taken into account in order to minimize total generation cost [1]. Also, appropriately reactive power flows relate to power losses and system voltage profiles, directly. In this viewpoint, tap setting of under-load tap-changing transformers (ULTCs), voltage magnitude of voltage-controlled buses or installing reactive power sources can improve voltage characteristics and reduce power losses, considerably [2]. In 1968, Dommel and Tinney [3] proposed the method of OPF, which employs allocation of real power generated by generators to co-operate with controlling ULTCs, magnitude of voltage-controlled buses or reactive power sources for minimizing the system objective.

As a general approach, a typical OPF problem employs an optimization method for balancing the power flow equations, and finding the optimum solution. The solution satisfies the constraint of the minimum value of an objective function, ie total generation cost for most cases, within the entire search space. The OPF problem is in general non-convex and non-linear. It may exist many local minima. Many mathematical techniques have been developed and applied to this problem such as linear programming, interior point method, quadratic programming, etc [4]. The algorithms essentially need some problem simplification such that the problem is linear or convex. Thus, a true global minimum cannot be guaranteed. Then, stochastic optimization methods such as Genetic Algorithms (GAs), Simulated Annealing (SA), Artificial Neural Network (ANN), Evolutionary Programming (EP), Particle Swarm Optimization (PSO) [5-9], were applied for solving such a problem directly without any simplification. These methods can successfully manipulate a problem in non-convex or non-linear. Therefore, an obtained optimal solution is more accurate and realistic. Unfortunately, these algorithms normally take lengthy calculation time when compared with the mathematical optimization methods. In 2005, artificial bees colony has been introduced and performed drastically improved search performance. Regarded as one of stochastic search processes, ABC provides a near global minimum by successfully avoiding local-minimum traps. Successful applications of the ABC have emerged, such as in [10-12]. Although the ABC algorithm was developed as a stochastic optimization technique, it can find an optimal solution within a short calculation time. In November 2009, Jing Qin introduced a key cutting algorithm that emulates the lock picking work of locksmiths to open a lock [13]. This algorithm is simple to understand and be implemented. In Qin's paper, a 9-number puzzle

Manuscript received on August 1, 2011 ; revised on October 14, 2011.

[1,2] The authors are with the Power System Research Unit, School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand. , E-mail: Uthen.leeton@gmail.com and Thanatchai@gmail.com

and a quadratic function of a single variable were used for test. The results were satisfactory but limited. In our further work, this key cutting algorithm always fails when the number of control variables is equal or greater than two. With respect to the original key cutting algorithm, some modifications are made in order to improve the performance of the algorithm suitable for unconstrained optimization problems.

This paper organizes a total of five sections. Next section, Section two, illustrates optimal power flow problems with corresponding mathematical expressions of its objective and various practical constraints. Section three gives the brief of three intelligences algorithm (GA, PSO, and KCA) for comparative purposes. It also provides the algorithm procedure, described step-by-step. Section four is the simulation results and discussion. Conclusion remark is in Section five.

2. OPTIMAL POWER FLOW PROBLEMS

2.1 Problem Formulation

The optimal power flow problem is a nonlinear optimization problem. It consists of a nonlinear objective function defined with nonlinear constraints. The optimal power flow problem requires the solution of nonlinear equations, describing optimal and/or secure operation of power systems. The general optimal power flow problem can be expressed as a constrained optimization problem as follows.

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{Subject to} && g(x) = 0, \text{ equality constraints} \\ & && h(x) \leq 0, \text{inequality constraints} \end{aligned}$$

By converting both equality and inequality constraints into penalty terms and therefore added to form the penalty function as described in (1) and (2).

$$P(x) = f(x) + \Omega(x) \tag{1}$$

$$\Omega(x) = \rho\{g^2(x) + [\max(0, h(x))]^2\} \tag{2}$$

Where $P(x)$ is the penalty function
$\Omega(x)$ is the penalty term
ρ is the penalty factor

By using a concept of the penalty method [14], the constrained optimization problem is transformed into an unconstrained optimization problem in which the penalty function as described above is minimized.

2.2 Objective Function

Although most of optimal power flow problems involve the total production cost of the entire power system, in some cases some different objective may be chosen. The objective function to be optimized is the total production cost. This function is a combination of all generators' fuel costs in the entire system. The fuel-cost characteristic of each generator is represented by using either a quadratic function. Minimization of total fuel cost. The generator cost curves are represented by quadratic functions as

$$\begin{aligned} F_T &= \sum_{i=1}^{N_G} f_i(P_{Gi}) \qquad (\text{B}/h) \\ &= \sum_{i=1}^{N_G} a_i + b_i P_{Gi} + c_i P_{Gi}^2 + |d_i \sin e_i (P_{Gi}^{min} - P_{Gi})| \end{aligned} \tag{3}$$

Where:
f_i is the fuel cost of the i^{th} generator
a_i, b_i, c_i, d_i and e_i are the cost coefficients of the i^{th} generator
P_{Gi} is the real power output of the i^{th} generator
N_G is the total number of generators

2.3 System Constraints

The controllable system quantities are generator MW, controlled voltage magnitude, reactive power injection from reactive power sources and transformer tapping. The objective use herein is to minimize the power transmission loss function by optimizing the control variables within their limits. Therefore, no violation on other quantities (e.g. MVA flow of transmission lines, load bus voltage magnitude, generator MVAR) occurs in normal system operating conditions. These are system constraints to be formed as equality and inequality constraints as shown below.

Equality constraint: These constraints represent load flow equations as:

$$P_{G,i} - P_{D,i} = \sum_{j=1}^{N_B} |V_i||V_j||Y_{i,j}| \cos(\theta_{i,j} - \delta_i + \delta_j) = 0 \tag{4}$$

$$Q_{G,i} - Q_{D,i} = \sum_{j=1}^{N_B} |V_i||V_j||Y_{i,j}| \sin(\theta_{i,j} - \delta_i + \delta_j) = 0 \tag{5}$$

Where: P_{Gi} is the real power generation at bus i
Q_{Gi} is the reactive power generation at bus i
P_{Di} is the real power demand at bus i
Q_{Di} is the reactive power demand at bus i
N_B is the total number of buses
$\theta_{i,j}$ is the angle of bus admittance element i, j
$Y_{i,j}$ is the magnitude of bus admittance element i, j

1) Inequality constraint: Variable limitations

$$V_i^{min} \le V_i \le V_i^{max} \tag{6}$$

$$T_i^{min} \le T_i \le T_i^{max} \tag{7}$$

$$Q_{comp,i}^{min} \le Q_{comp,i} \le Q_{comp,i}^{max} \tag{8}$$

$$P_{G,i}^{min} \le P_{G,i} \le P_{G,i}^{max} \tag{9}$$

Where: V_i^{min}, V_i^{max} are upper and lower limits of voltage magnitude at bus i
T_i^{min}, T_i^{max} are upper and lower limits of tap position of transformer i
$Q_{comp,i}^{min}, Q_{comp,i}^{max}$ are upper and lower limits of reactive power source i
$P_{G,i}^{min}, P_{G,i}^{max}$ are upper and lower limits of power generated by generator i

2.4 Control Parameters

Initially, the KCA was designed to work on a binary-valued encoding representation of the problem parameters. During the searching process, a locksmith as a collection of the key set represents a solution vector (keys). Let $\bar{x}_i$ be a created key set as possible solutions. The power output of all generating units $(\vec{P}, \vec{Q})$, the voltage magnitude and angle of all buses $(|\vec{V}|, \vec{\delta})$ and the tapping value of all transformers $(\vec{T})$ and reactive power injection from reactive power compensators $(\vec{Q}_c)$ are typical members of the solution vector and it can be written as described in (10).

$$\vec{x}_i = \begin{bmatrix} \vec{P} & \vec{Q} & |\vec{V}| & \vec{\delta} & \vec{T} & \vec{Q}_C \end{bmatrix}^T \tag{10}$$

2.5 Control Parameters

The total production cost is computed as the sum of the individual unit costs and therefore used as the system objective function. To account for all the system constraints (12) - (17), the total production cost is augmented by non-negative penalty terms to penalize the constraint violations. Thus, the augmented cost function, called the penalty function [15-16], is formed as (11).

$$P(x) = F_T + \Omega_P + \Omega_Q + \Omega_C + \Omega_T + \Omega_V + \Omega_G \tag{11}$$

Where:

$$\Omega_P = \rho \sum_{i=1}^{N_B} \left\{ P_{G,i} - P_{D,i} - \sum_{j=1}^{N_B} |V_i||V_j||Y_{ij}| \cos(\theta_{i,j} - \delta_i + \delta_j) \right\}^2 \tag{12}$$

$$\Omega_Q = \rho \sum_{i=1}^{N_B} \left\{ Q_{G,i} - Q_{D,i} + \sum_{j=1}^{N_B} |V_i||V_j||Y_{ij}| \sin(\theta_{i,j} - \delta_i + \delta_j) \right\}^2 \tag{13}$$

$$\Omega_C = \rho \sum_{i=1}^{N_C} \left\{ \max(0, Q_{comp,i} - Q_{comp,i}^{\max}) \right\}^2 + \rho \sum_{i=1}^{N_C} \left\{ \max(0, Q_{comp,i}^{min} - Q_{comp,i}) \right\}^2 \tag{14}$$

$$\Omega_T = \rho \sum_{i=1}^{N_C} \{\max(0, T_i - T_i^{\max})\}^2 + \rho \sum_{i=1}^{N_C} \{\max(0, T_i^{\min} - T_i)\}^2 \tag{15}$$

$$\Omega_V = \rho \sum_{i=1}^{N_B} \{\max(0, V_i - V_i^{\max})\}^2 + \rho \sum_{i=1}^{N_B} \{\max(0, V_i^{\min} - V_i)\}^2 \tag{16}$$

$$\Omega_G = \rho \sum_{i=1}^{N_G} \{\max(0, P_{G,i} - P_{G,i}^{\max})\}^2 + \rho \sum_{i=1}^{N_G} \{\max(0, P_{G,i}^{\min} - P_{G,i})\}^2 \tag{17}$$

N_G is the total number of generators
N_C is the total number of reactive power sources
N_T is the total number of transformers

3. INTELLIGENT SEARCH METHODS

3.1 Genetic Algorithm (GA)

There exist many different approaches to adjust the control parameters. The GA is well-known [17] there exist a hundred of works employing the GA technique to optimize the system objective in various forms. The GA is a stochastic search technique that leads a set of population in solution space evolved using the principles of genetic evolution and natural selection, called genetic operators e.g. crossover, mutation, etc. With successive updating new generation, a set of updated solutions gradually converges to the real solution. Because the GA is very popular and widely used in most research areas where an intelligent search technique is applied, it can be summarized briefly as shown in the flowchart in figure 1 [18].

In this paper, the GA is selected to build up an algorithm to solve optimal power flow problems (all generation from available generating units). To reduce programming complication, the Genetic Algorithm (GADS TOOLBOX in MATLAB [19]) is employed to generate a set of initial random parameters. With the searching process, the parameters are adjusted to give the best result.

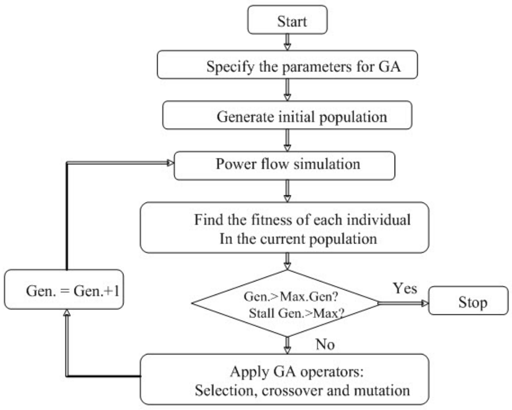

Fig.1: *Flowchart of the GA procedure*

3.2 Particle Swarm Optimization (PSO)

Kennedy and Eberhart developed a particle swarm optimization algorithm based on the behavior of individuals (i.e., particles or agents) of a swarm [20]. Its roots are in zoologist's modeling of the movement of individuals (i.e., fish, birds, and insects) within a group. It has been noticed that members of the group seem to share information among them to lead to increased efficiency of the group. The particle swarm optimization algorithm searches in parallel using a group of individuals similar to other AI-based heuristic optimization techniques. Each individual corresponds to a candidate solution to the problem. Individuals in a swarm approach to the optimum through its present velocity, previous experience, and the experience of its neighbors. In a physical n-dimensional search space, the position and velocity of individual i are represented as the velocity vectors. Using these information individual i and its updated velocity can be modified under the following equations in the particle swarm optimization algorithm. The procedure of the particle swarm optimization can be summarized in the flow diagram of figure 2.

$$x_i^{k+1} = x_i^{(k)} + v_i^{k+1} \tag{18}$$

$$v_i^{k+1} = v_i^{(k)} + \alpha_i \left(x_i^{lbest} - x^{(k)_i} \right) + \beta_i \left(x^{gbest} - x_i^{(k)} \right) \tag{19}$$

Where
$x_i^{(k)}$ is the individual i at iteration k
$v_i^{(k)}$ is the updated velocity of individual i at iteration k
α_i, β_i are uniformly random numbers between [0,1]
x_i^{lbest} is the individual best of individual i
x^{gbest} is the global best of the swarm

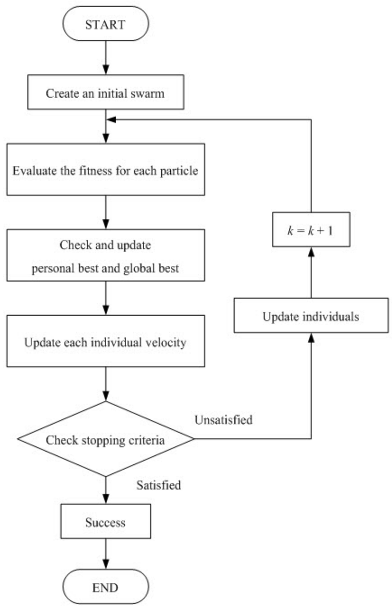

Fig.2: *Flowchart of the PSO procedure*

3.3 Key Cutting Algorithm (KCA)

Key cutting algorithm (KCA) is a new algorithm introduced by Jing Qin in 2009 [13]. It is based on the work of a locksmith to find a perfect match of the lost key that can open the lock. People often encounter a situation that they lost keys in their daily life and when this happens they can turn to a locksmith to help. One possible solution is that the locksmith chooses multiple similar types of keys and tries to open the lock, and see which keys are much easier to get into the lock hole. He then finds similarities among those keys and based on those similarities he makes another set of possible keys and repeats the same procedure until he has a winner - a key opens the lock. To make such problem more general, an objective function of any problem can be considered as a lock without key and a solution to the problem is a key to open the lock. Similar procedures as the locksmith does can be employed to solve the optimization problems. This algorithm can be described as follows.

A.Definitions

Definition 1 Lock: Problem requires solution. For this example the lock is Find the X when Y has the minimum value.

Definition 2 Key: One possible solution for the problem. For the quadratic equation problem, A real number for X, like X = 4 is a key.

Definition 3 Tooth: One component of the vector of the solution. One tooth on the key is one component of the vector of the solution. Here, X is a real value; we can encode a real value with its binary format. X=5 (binary is 00000101)

Fig.3: *One component of the vector of the solution*

Definition 4 Key Set: A set of possible solutions.

Definition 5 Fitness: The degree of how the key and lock matches. The differences between the candidate key and the key open the lock.

Definition 6 Similarity: The degree of similarities among all keys in a key set.

Definition 7 Key cutting: Adjust one tooth on the key or change one component in the solution vector. For instance, X=5 (0000101B), change the 2nd tooth from 0 to 1, 0101⇒0111.

Definition 8 Probability factor of Key Cutting: The probability uses to control the variation of one tooth. For instance, X=5 (00000101B). This probability is used to control each 1 and 0 in 00000101 the probability of changing from 1 to 0 or vice versa. In the key cutting algorithm, the probability factor of key cutting is calculated based on the similarity in the key set.

Definition 9 Selection: Choose a subset from a key set to go into next iteration.

B. Basic Algorithm procedure

Assume a key is k= [s_n, s_{n-1},. . .,s_1]

Step 1: Encoding the Key.

Step 2: Random generate the initial Key Set K_0, total number of keys is $2m$.

Step 3: Calculate the Fitness of each Key in the Key set.

Step 4: Selection half of all Keys in the Key Set with a higher Fitness, which makes a new Key Set K_1'

Step 5: Calculate the Probability factor of Key Cutting

$p_{ij}(i = 1, 2, m; j = 1, 2, n)$ for each Tooth S_{ij}.

Use a matrix, the key vector in Key Set K_1':

$$K' = \begin{bmatrix} S_{1n} & S_{1(n-1)} & \cdots & S_{12} & S_{11} \\ S_{2n} & S_{2(n-1)} & \cdots & S_{22} & S_{21} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{(m-1)n} & S_{(m-1)(n-1)} & \cdots & S_{(m-1)2} & S_{(m-1)1} \\ S_{mn} & S_{m(n-1)} & \cdots & S_{m2} & S_{m1} \end{bmatrix}$$

p_{ij} =1-(the number of S_{ij} in column J/m)

Step 6: Based on p_{ij}, perform the Key Cutting for each Tooth S_{ij} in Key Set K_1'. And create a new Key Set K_1". K_1' and K_1" together makes up of Key Set K_1

Step 7: Repeat 2 to 5 until one of the following condition is met, output the best Key in the final Key Set as the final solution.

Termination condition:

1) reaches the maximum iteration.
2) within predefined solution variance.
3) All Keys are the same in the Key Set.

The key cutting algorithm can be summarized in the following steps or as shown in figure 4.

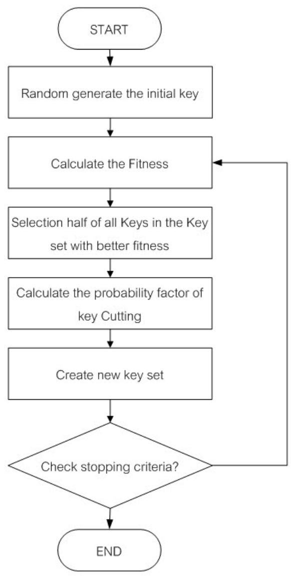

Fig.4: *: Flowchart of the KCA procedure*

C. Modifications of the Key Cutting Algorithm

In the original key cutting algorithm, it converges to a solution very fast. However, in multivariate problems, it usually fails to find the optimal solution. A key set with key cutting probability is used and key picking is repeatedly performed iteration-byiteration, until all keys are the same. The major disadvantage of this algorithm occurs when a majority of keys in a key set is not good enough to open the lock. The similarity acquired among those keys can cause a trap of solutions to force all keys in a key set being the same. To avoid such a trap, additional feature must be inserted to create various candidates to open the lock. In this paper, four strategies are tried as follows.

Modification 1 (KCA1): Tooth adjustment is performed at only one tooth with the highest key cutting probability and the highest probability must be greater than 0.5, otherwise skip the adjustment. However, there might be more than one tooth having the same value of the highest probability, only one tooth will be selected randomly for simplification.
Modification 2 (KCA2): Similar to the modification 1. To increase opportunity of finding a good key candidate, all teeth having the same value of the highest probability will be performed the tooth adjustment. Only one adjustment that gives the best fitness is selected to create a new key instead of using the random selection as applied in Modification 1.
Modification 3 (KCA3): Tooth adjustment is applied to all teeth having the probability higher than 0.5. Only one adjustment that gives the best fitness is selected to create a new key.
Modification 4 (KCA4): Similar to the modification 3. However, if the key cutting probability of a new key set is equal to the key cutting probability of the key set from the previous iteration, then generate a new key set and restart the process.

4. RESULTS AND DISCUSSION

The test systems were separated into two groups. The first group consisted of four standard mathematical functions test of small-scale. The second group employed three standard IEEE test power systems which were the 6-bus, 14-bus and 30-bus test power systems [21]. The detail of each group was given as follows.

4.1 Mathematical Test Functions

This paper uses the proposed algorithm to minimize the following two typical multi-peak functions which has many local extreme values. During the process of optimization, it is easy to become stuck in these local minima and lead to a premature convergence. To compare results obtained by using genetic algorithm and key cutting algorithm, all test cases were simulated by using the same computer which was an Intel®, Core 2 Duo, 2.4 GHz, 3.0 GB RAM.

4.1.1 Test Function 1

The test of this function is carried out by applying the same parameter setting to all the key cutting algorithms and genetic algorithms as follows.

- Population size is 80
- Maximum iteration is 50
- No stalled generation is applied
- 16-bit resolution is used for each variable

$$f(x_1, x_2) = -\frac{\sin(x_1)}{x_1} \times \frac{\sin(x_2)}{x_2} \quad (20)$$

For the test functions 1 shown as equation (20) [22] has infinitely local minima. The point (0,0) is said to be the global minimum. The objective at its global minimum is 0 as the 3D characteristic of the test function 1 shown in figure 5. In table 1 and 2 are fitness value solution, average CPU computational time and an optimal solution adjustment for each method. Comparisons of the convergence among GA and KCA for the test function one were presented in figure 6.

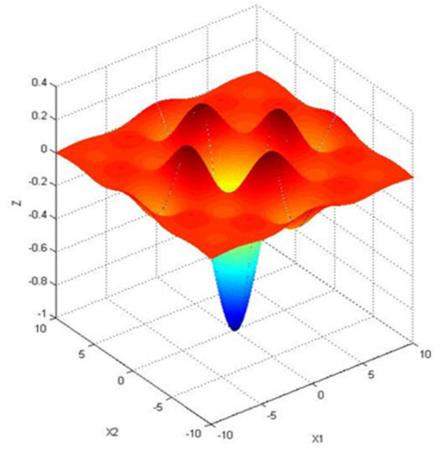

Fig.5: *3D characteristic of the test function 1*

After 30 trials of solutions, the selected convergence from each method is shown in figure 6. KCA2 and KCA4 are the two best methods for finding

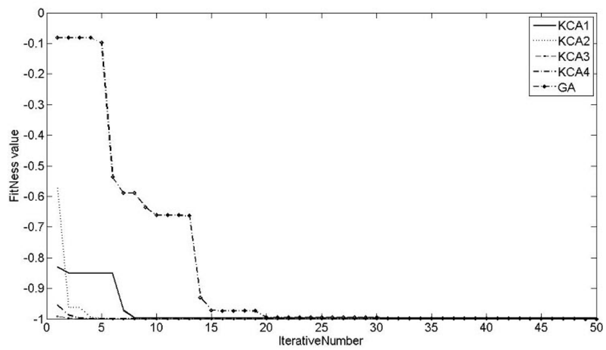

Fig.6: *Convergences of the test function 1*

Table 1: *Statistic of fitness value and average CPU computational time for the test function 1*

Method	Statistic of fitness value				Average CPU Time (s)
	Minimum	Maximum	Average	Deviation	
KCA 1	-0.9992	-0.5400	-0.8582	0.1357	0.8730
KCA 2	-1.0000	-1.0000	-1.0000	0.0000	8.1380
KCA 3	-1.0000	-0.9222	-0.9959	0.0143	0.9211
KCA 4	-1.0000	-1.0000	-1.0000	0.0000	3.1484
GA	-1.0000	-0.9999	-1.0000	0.0000	0.4692

Table 2: *Statistic of optimal solution adjustment for the test function 1*

Method	Variable	Statistic value of adjustment			
		Minimum	Maximum	Average	Deviation
KCA 1	X1	-1.5334	1.4020	-0.0514	0.7756
	X2	-1.5462	1.4424	0.1240	0.5975
KCA 2	X1	0.0000	0.0000	0.0000	0.0000
	X2	0.0000	0.0000	0.0000	0.0000
KCA 3	X1	-0.1791	0.2948	0.0045	0.0879
	X2	-0.0501	0.6275	0.0428	0.1269
KCA 4	X1	-0.0062	0.0091	-0.0001	0.0022
	X2	-0.0030	0.0012	-0.0002	0.0009
GA	X1	-0.0179	0.0083	-0.0019	0.0071
	X2	-0.0148	0.0221	-0.0003	0.0070

4.1.2 Test Function 2

The test of this function is carried out by applying the same parameter setting to all the key cutting algorithms and genetic algorithms as follows.

- Population size is 80
- Maximum iteration is 50
- No stalled generation is applied
- 16-bit resolution is used for each variable

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \qquad (21)$$

For the test functions 2 shown as equation (21) has an infinitely local minimum. The point (1,1) is said to be the global minimum. The objective at its global minimum is 0 as the 3D characteristic of the test function 2 shown in figure 7. In Table 3 and 4 are fitness value solution, average CPU computational time and an optimal solution adjustment for each method. Comparisons of the convergence among GA and KCA for the test function one were presented in figure 8.

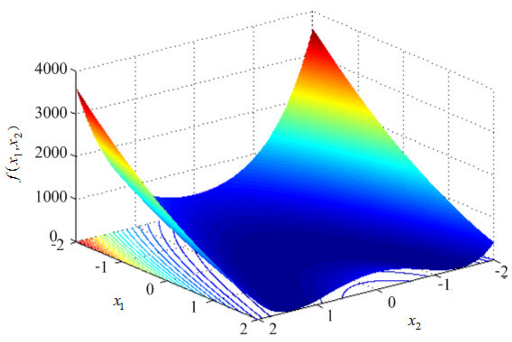

Fig.7: *3D characteristic of the test function 2*

After 30 trials of solutions, the selected convergence from each method is shown in figure 8. KCA4 is the best method for finding the best objective function, while GA is the fastest.

4.1.3 Test Function 3

The test of this function is carried out by applying the same parameter setting to all the key cutting algorithms and genetic algorithms as follows.

- Population size is 80
- Maximum iteration is 100
- No stalled generation is applied
- 20-bit resolution is used for each variable

$$f(x_1, x_2, x_3) = 30 + \sum_{i=1}^{3} \left(\frac{1}{10}(x_i - 20)^2 - 9\cos(\frac{2\pi x_i}{5}) \right) \qquad (22)$$

For the test functions 3 shown as equation (22) has an infinitely local minimum. The point (20,20,20) is said to be the global minimum. The objective at its global minimum is 3 as the 3D characteristic of the test function 3 shown in figure 9. In table 5 and 6 are fitness value solution, average CPU computational time and an optimal solution adjustment for each method. Comparisons of the convergence among GA and KCA for the test function one were presented

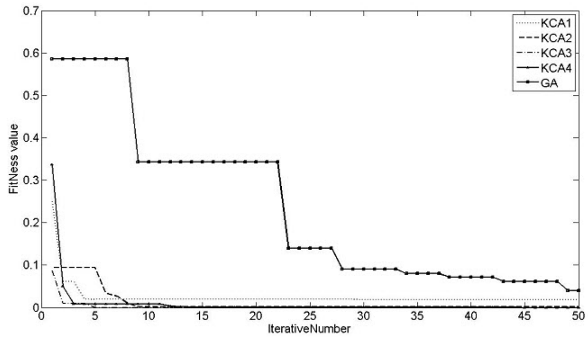

Fig.8: *Convergences of the test function 2*

Table 3: *Statistic of fitness value and average CPU computational time for the test function 2*

Method	Statistic of fitness value				Average CPU Time (s)
	Minimum	Maximum	Average	Deviation	
KCA 1	0.0004	2.3670	0.3416	0.5410	1.9529
KCA 2	0.0015	2.8507	0.2101	0.5155	14.4581
KCA 3	0.0000	0.0694	0.0081	0.0149	1.7735
KCA 4	0.0000	0.0002	0.0000	0.0001	7.3794
GA	0.0000	0.0003	0.0001	0.0001	1.1487

Table 4: *Statistic of optimal solution adjustment for the test function 2*

Method	Variable	Statistic value of adjustment			
		Minimum	Maximum	Average	Deviation
KCA 1	X1	0.5287-	1.3932	0.8152	0.4145
	X2	0.0895-	2.0104	0.8276	0.5479
KCA 2	X1	0.6839-	1.3381	0.7832	0.4087
	X2	0.0640	1.7904	0.7739	0.5143
KCA 3	X1	0.7809	1.1141	0.9768	0.0793
	X2	0.6244	1.2417	0.9609	0.1480
KCA 4	X1	0.9853	1.0081	0.9996	0.0048
	X2	0.9704	1.0162	0.9993	0.0095
GA	X1	0.9818	1.0166	0.9989	0.0085
	X2	0.9642	1.0339	0.9978	0.0172

in figure 10.

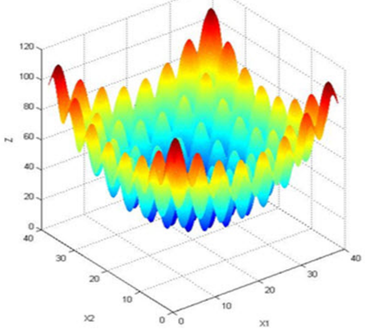

Fig.9: *3D characteristic of the test function 3*

After 30 trials of solutions, the selected convergence from each method is shown in figure 10. KCA4 and GA are the two best methods for finding the best objective function, while only GA is the fastest.

Table 5: *Statistic of fitness value and average CPU computational time of the test function 3*

Method	Statistic of fitness value				Average CPU Time (s)
	Minimum	Maximum	Average	Deviation	
KCA 1	7.7601	26.5507	15.5556	4.7143	5.1233
KCA 2	3.0000	8.3936	3.9888	1.4996	65.4959
KCA 3	3.0000	6.7411	3.9021	1.1850	4.6791
KCA 4	3.0000	3.0047	3.0007	0.0010	27.2741
GA	3.0000	3.0024	3.0003	0.0004	2.4452

Table 6: *Statistic of optimal solution adjustment of the test function 3*

Method	variable	Statistic value of adjustment			
		Minimum	Maximum	Average	Deviation
KCA 1	X_1	10.4113	29.4252	19.4668	4.0895
	X_2	9.9498	29.9537	20.3695	5.0867
	X_3	13.8850	29.8439	20.9554	4.7510
KCA 2	X_1	15.2500	24.7500	19.8417	1.5192
	X_2	15.2500	24.7500	19.6833	2.1364
	X_3	20.0000	24.7500	20.3167	1.2051
KCA 3	X_1	19.8437	20.3276	20.0234	0.1226
	X_2	15.0688	20.2973	19.8392	0.9074
	X_3	19.4061	24.9652	20.8266	1.8008
KCA 4	X_1	19.9903	20.0124	20.0003	0.0041
	X_2	19.9784	20.0103	19.9979	0.0076
	X_3	19.9912	20.0175	20.0006	0.0051
GA	X_1	19.9933	20.0066	20.0003	0.0028
	X_2	19.9946	20.0097	20.0006	0.0033
	X_3	19.9819	20.0074	19.9993	0.0051

4.1.4 Test Function 4

The test of this function is carried out by applying the same parameter setting to all the key cutting algorithms and genetic algorithms as follows.

- Population size is 30
- Maximum iteration is 50

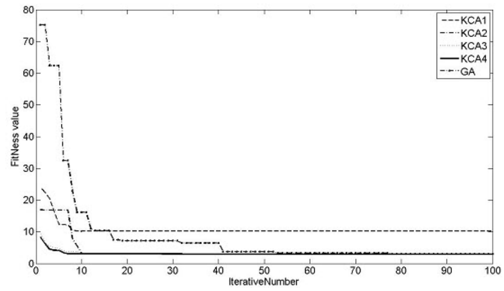

Fig.10: *Convergences of the test function 3*

- No stalled generation is applied
- 16-bit resolution is used for each variable

For the test functions 4 shown as equation (23) has an infinitely local minimum. The point (0,0) is said to be the global minimum. The objective at its global minimum is 0 as the 3D characteristic of the test function 4 shown in figure 11. In table 7 and 8 are fitness value solution, average CPU computational time and an optimal solution adjustment for each method. Comparisons of the convergence among GA and KCA for the test function one were presented in figure 12.

$$f(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001 * (x_1^2 + x_2^2))^2} \tag{23}$$

For the test functions 4 shown as equation (23) has an infinitely local minimum. The point (0,0) is said to be the global minimum. The objective at its global minimum is 0 as the 3D characteristic of the test function 4 shown in figure 11. In table 7 and 8 are fitness value solution, average CPU computational time and an optimal solution adjustment for each method. Comparisons of the convergence among GA and KCA for the test function one were presented in figure 12.

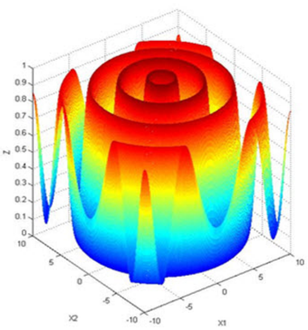

Fig.11: *3D characteristic of the test function 4*

After 30 trials of solutions, the selected convergence from each method is shown in figure 12. KCA4 is the best method for finding the best objective function, while GA is the fastest.

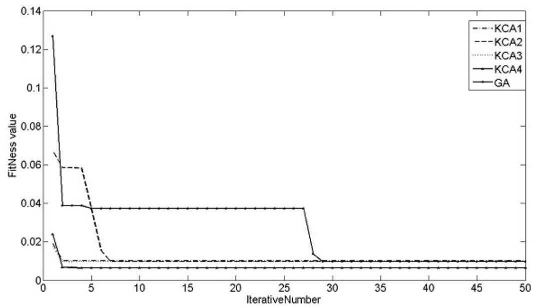

Fig.12: *Convergences of the test function 4*

Table 7: *Statistic of fitness value and average CPU computational time of the test function 4*

Method	Statistic of fitness value				Average CPU Time (s)
	Minimum	Maximum	Average	Deviation	
KCA 1	0.0097	0.0642	0.0175	0.0128	7.8100
KCA 2	0.0000	0.0372	0.0103	0.0054	69.8172
KCA 3	0.0003	0.0099	0.0088	0.0028	6.7668
KCA 4	0.0000	0.0001	0.0000	0.0000	32.6587
GA	0.0000	0.0097	0.0006	0.0025	4.3281

Table 8: *Statistic of optimal solution adjustment of the test function 4*

Method	Variable	Statistic value of adjustment			
		Minimum	Maximum	Average	Deviation
KCA 1	X1	-3.1190	5.8280	0.4626	2.3785
	X2	-5.6602	5.9409	-0.4754	2.7057
KCA 2	X1	-3.0535	5.1441	0.9265	1.9840
	X2	-3.5986	3.1385	0.0055	2.4600
KCA 3	X1	-3.1318	3.0820	-0.5423	2.0219
	X2	-3.1510	3.1182	-1.0203	1.9247
KCA 4	X1	-0.0069	0.0051	-0.0005	0.0027
	X2	-0.0099	0.0081	-0.0005	0.0024
GA	X1	-0.0011	2.6667	0.1752	0.6669
	X2	-1.7724	0.0015	-0.1144	0.4350

4.2 IEEE Standard Test Systems

Although a modern electric power system consists of many types of power plants, in this research the tests focused on fossil power generation units only. A simple model of such a generator is made from its input fuel cost in /h and corresponding power output generated in MW as input and output variables, respectively [18]. The test case scenarios were divided into two main conditions. There were smooth and non-smooth fuel-cost curve conditions. In the smooth curve case, all generators' fuel-cost curves were quadratic whilst the second test employed a valve-point loading function [23,24]. Fuel cost function of each generator connected to the system is generally given in the form of valve-point loading function with having a quadratic term in it.

4.2.1 Smooth fuel cost case - quadratic cost function

In this case, a quadratic fuel-cost function was assigned to all generators. The test was performed on an Intel®, Core 2 Duo, 2.4 GHz, 3.0 GB RAM with MATLAB. The SQP and GA are used in these tests is of MATLAB Optimization Toolbox [25-26]. Other two methods were coded by using MATLAB programming. Three standard IEEE test power systems (6-bus, 14-bus and 30-bus) as shown in figure 13 - 15 were used for test. Variable limits given in table 9 to table 11 were used as system constraints for 6-bus, 14-bus and 30-bus test system respectively. For comparison purposes, sequential quadratic programming, genetic algorithm, particle swarm optimization, and Key cutting algorithm were applied to solve the test system with various cases. Each method was challenged by solving given optimal power flow problems of 30 trials randomly. Minimum, average, maximum and standard deviation of the 30 trial solutions for all test system obtained by each method were evaluated and shown in table 12. Table 13 showed the comparison of CPU time spent by each approach. The optimal control variables obtained by each method and all test system were shown in tables 14 - 16. In addition, the solution convergences for three selected test cases were shown in figure 16 - 18.

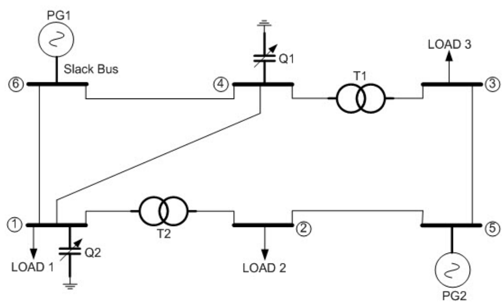

Fig.13: *Standard IEEE 6-bus test power system*

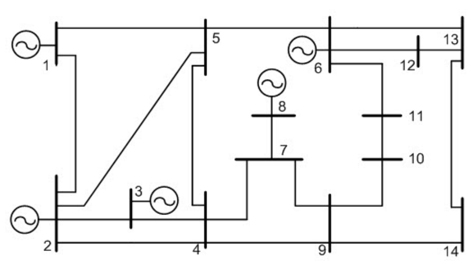

Fig.14: *Standard IEEE 14-bus test power system*

The results for the 6-bus test system showed that the KCA-based optimal power flow method gave the best results when compared with those obtained by the GA, PSO and the SQP. test system the average production cost solutions are 440.7 ฿/h, 470.8 ฿/h,

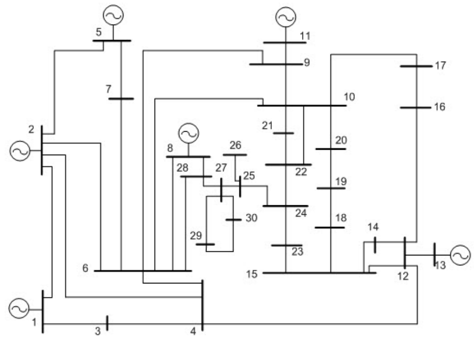

Fig.15: *Standard IEEE 30-bus test power system*

Table 9: *Variable limits and cost coefficients used for the 6-bus test system with smooth fuel cost function*

Variable	Limit		cost coefficients		
	Min.	Max.	a	b	c
MagV1 to V2 (p.u.)	1.0,1.1	1.1,1.15	-	-	-
Ang.V2 to V6 (◦)	-30	30	-	-	-
T1 to T2 (p.u.)	0.90	1.10	-	-	-
QC1 to QC2 (MVAR)	0,0	15,30	-	-	-
PG1(MW), QG1(MVAR)	50,-20	200,100	0	100	100
PG2(MW), QG2(MVAR)	20,-20	80,100	0	500	100

Table 10: *Variable limits and cost coefficients used for the 14-bus test system with smooth fuel cost function*

Variable	Limit		cost coefficients		
	Min.	Max.	a	b	c
Mag. V1 to V14 (p.u.)	0.95	1.10	-	-	-
Ang. V2 to V14 (degree)	-30	30	-	-	-
T1 to T5 (p.u.)	0.90	1.10	-	-	-
PG1(MW), QG1(MVAR)	50,-20	200,150	0.0150	2.45	150
PG2(MW), QG2(MVAR)	20,-20	80,60	0.0225	3.51	44.4
PG3(MW), QG3(MVAR)	15,-15	50,44.7	0.0175	2.75	55
PG4(MW), QG4(MVAR)	10,-15	30,62.5	0.013	3.89	40.6
PG5(MW), QG5(MVAR)	10,-10	35,40	0.0275	2.85	75

Table 11: *Variable limits and cost coefficients used for the 30-bus test system with smooth fuel cost function*

Variable	Limit		cost coefficients		
	Min.	Max.	a	b	c
Mag.V1 to V30 (p.u.)	0.95	1.10	-	-	-
Ang.V2 to V30 (degree)	-30	30	-	-	-
T1 to T4 (p.u.)	0.90	1.10	-	-	-
QG1 to QG9 (MVAR)	0	5	-	-	-
PG1(MW),QG1(MVAR)	50,-20	200,150	0.00375	2.00	0
PG2(MW),QG2(MVAR)	20,-20	80,60	0.01750	1.75	0
PG5(MW),QG5(MVAR)	15,-15	50,44.7	0.06250	0.06250	0
PG8(MW),QG8(MVAR)	10,-15	35,62.5	0.00834	3.25	0
PG11(MW),QG11(MVAR)	10,-10	30,40	0.02750	3.00	0
PG13(MW),QG13(MVAR)	12,-15	40,48.7	0.02500	3.00	0

474.2 B/h and 455.5 B/h for the KCA, GA, PSO and SQP methods, respectively.

Table 12: *OPF results obtained by using SQP, PSO, GA and KCA for all test system with smooth fuel cost function*

Test System	Method	Optimal total generaation costs (฿/h)			
		Minimum	Maximum	Average	Deviation
6 bus	SQP	354.6	621.7	455.5	71.1
	PSO	387.0	591.4	474.2	56.8
	GA	387.0	594.0	470.8	66.9
	KCA	396.5	526.4	440.7	34.2
14 bus	SQP	1184.7	1682.5	1388.4	159.2
	PSO	1285.5	1777.4	1507.3	122.7
	GA	1386.6	1781.5	1613.4	122.3
	KCA	1358.1	1786.3	1546.5	89.00
30 bus	SQP	629.2	12160.0	1396.9	2172.3
	PSO	639.7	847.6	720.8	53.0
	GA	708.9	896.4	802.5	47.6
	KCA	735.0	937.6	826.7	50.3

The result of the 14-bus system, SQP is the best solution at 1388.4 B/h. However, when the total number of buses is large, the KCA is limited. The SQP can find the best among the four competitors while the KCA came the third but deviation of KCA is the best solution at 89.0 B/h.

Table 13: *Computational time to using SQP, PSO, GA and KCA for all test system with smooth fuel cost function*

Test System	Method	Computational Time Consumed (s)			
		Minimum	Maximum	Average	Deviation
6 bus	SQP	0.24	15.84	4.14	4.3759
	PSO	8.40	8.75	8.49	0.0639
	GA	8.95	9.46	9.05	0.1290
	KCA	2.96	4.16	3.59	0.3191
14 bus	SQP	5.62	22.61	10.08	4.09
	PSO	55.78	107.32	86.59	22.44
	GA	67.94	72.74	69.94	1.63
	KCA	38.67	85.47	61.56	12.63
30 bus	SQP	73.5	200.1	126.4	32.61
	PSO	2130.6	3047.2	2948.2	237.16
	GA	2242.3	2251.1	2245.7	2.25
	KCA	1062.6	1843.8	1524.4	186.86

The result of the 30-bus system is similar to that of the 14-bus system However, when the total number of buses is large, the KCA is limited. The PSO can find the best among the four competitors while the KCA came the third. These simulation results reveal that the KCA method is very effective in finding the optimum of the optimal power flow problems, especially with a small-scale test power system. The CPU times spent by each method to find the optimal solution showed that the KCA consumed the least computa-

Table 14: *Optimal solution by each method for the 6-bus system with smooth fuel cost function*

Variable	PSO	GA	KCA
Gen1 (MW),(MVAR)	114.12, 39.71	125.37, 32.90	125.29, 54.82
Gen2 (MW), (MVAR)	30.33, 34.05	20.07, 73.43	21.88, -7.76
Mag_V1 (p.u.)	1.0855	1.0388	1.0827
Mag_V2 (p.u.)	1.1263	1.1003	1.1131
QC1(MVAR)	7.68	4.06	11.29
QC2(MVAR)	16.61	27.70	21.41
T1(T3-4) (p.u.)	1.0433	0.9000	0.9620
T2(T5-6) (p.u.)	0.9595	0.9996	0.9573
PG total (MW)	144.45	145.44	147.17

Table 15: *Optimal solution by each method for the 14-bus system with smooth fuel cost function*

variable	PSO	GA	KCA
Gen1 (MW),(MVAR)	137.4,130.7	132.4,131.7	138.2,130.0
Gen2 (MW), (MVAR)	64.0, -11.1	71.3, 8.2	52.4,13.8
Gen3 (MW),(MVAR)	37.4, 18.6	41.5, 23.3	50,3.2
Gen6 (MW),(MVAR)	22.9, 39.7	24.7, 3.0	18.7,10.8
Gen8 (MW),(MVAR)	24.0, 21.8	16.6, 32.5	28.3,14.5
MagV1 (p.u.),AngV1 (◦)	1.0638,0	1.001,0	0.9606,0
MagV2 (p.u.),AngV2 (◦)	0.9849,-5.48	1.0432,1.48	0.9829,-4.35
MagV3 (p.u.), AngV3 (◦)	1.0533,-10.59	1.0319,-9.14	1.0329,-5.76
MagV6 (p.u.), AngV6 (◦)	1.0489,-12.55	0.9893,11.15	1.0618,-0.11
MagV8 (p.u.), AngV8 (◦)	0.9657,-0.73	0.9728,12.20	1.1000,12.58
T1 (T4-7) (p.u.)	0.9862	0.9594	0.9235
T2 (T4-9) (p.u.)	1.0257	1.0137	0.9463
T3 (T5-6) (p.u.)	0.9847	1.0155	0.9431
T4 (T7-8) (p.u.)	0.9737	1.0997	0.9149
T5 (T7-9) (p.u.)	0.9718	0.9616	1.0773
PG total (MW)	273	287	287

tional time effort for 6 bus test system, However SQP consumed the least computational time effort for 14 and 30 bus test system.

Table 16: *Optimal solution by each method for the 30-bus system with smooth fuel cost function*

Variable	PSO	GA	KCA
Gen1 (MW),(MVAR)	85.7, 110.5	154.3,83.4	158.2,132. 0
Gen2 (MW) (MVAR)	57.0, 26.7	29.2, 50.0	30.5, 47.4
Gen5 (MW),(MVAR)	19.2, 12.4	24.2,36.4	26.8, 35.5
Gen8 (MW),(MVAR)	28.3, 26.0	19.6,49.7	23.6, 55.5
Gen11 (MW),(MVAR)	21.6, 15.8	10.6, 36.1	14.0, 34.9
Gen13 (MW),(MVAR)	18.8, 18.8	22.8, 43.8	15.9, 34.7
MagV1 (p.u.),AngV1 (◦)	1.0171,0	0.9640,0	1.0024,0
MagV2 (p.u.),AngV2 (◦)	1.0088, -0.18	0.9738,-3.93	1.0053, 1.52
MagV5 (p.u.),AngV5 (◦)	1.0497, -4.25	1.0546,-10.61	1.0312, 1.05
MagV8 (p.u.),AngV8 (◦)	0.9856, 8.17	1.0152,-3.56	0.9506, 0.58
MagV11 (p.u.),AngV11 (◦)	1.0516, 1.78	1.0520, 4.42	0.9847, -26.70
MagV13 (p.u.),AngV13 (◦)	1.0458, -1.51	1.0876, 9.86	0.9812, 6.00
T1 (T6-9) (p.u.)	1.0998	0.9938	1.0333
T2 (T6-10) (p.u.)	1.0489	1.0966	1.0090
T3 (T4-12) (p.u.)	1.0696	1.0580	0.9612
T4(T25-27) (p.u.)	1.0504	1.0104	0.9612
QC10(MVAR)	1.5179	3.5526	3.3137
QC12(MVAR)	3.8267	2.4982	4.5098
QC15(MVAR)	1.6685	3.4170	3.5294
QC17(MVAR)	2.1742	3.6650	4.4706
QC20(MVAR)	3.7568	1.5134	4.9804
QC21(MVAR)	1.2999	4.5610	4.7451
QC23(MVAR)	0.8187	3.4497	4.6275
QC24(MVAR)	1.3257	4.5451	0.4118
QC29(MVAR)	3.2454	3.3406	4.8824
PG total (MW)	212	238.1	253.3

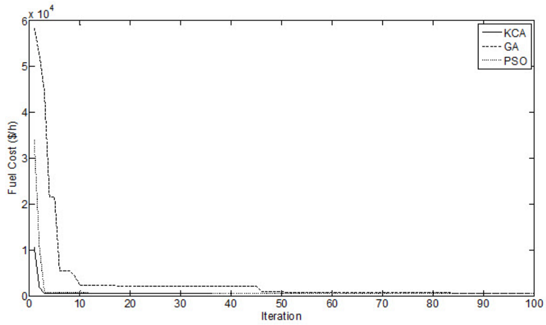

Fig.16: *Convergence characteristics of each method for the 6-bus test system with smooth fuel cost function*

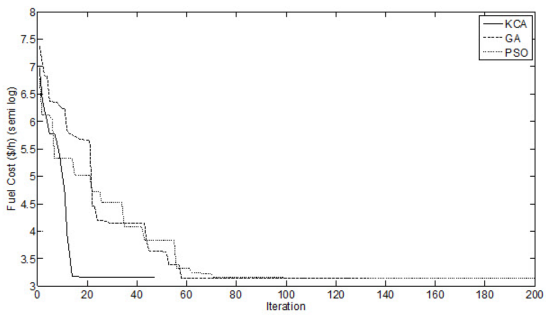

Fig.17: *Convergence characteristics of each method for the 14-bus test system with smooth fuel cost function*

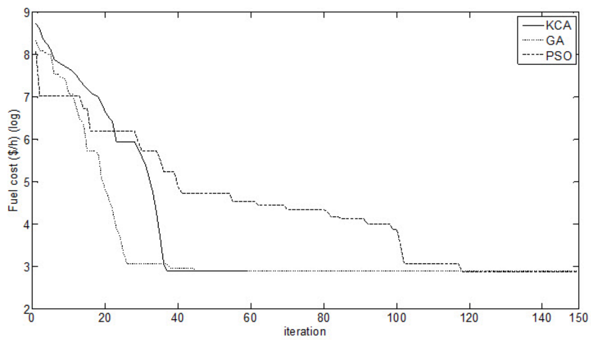

Fig.18: *Convergence characteristics of each method for the 30-bus test system with smooth fuel cost function*

4.2.2 Non-smooth fuel cost case - valve point loading

In this case, a valve-effect fuel-cost function was assigned to all generators. Three standard IEEE test power systems (6-bus, 14-bus and 30-bus) were used for test. Variable limits given in table 16 to table 18 were used as system constraints for 6-bus, 14-bus and 30-bus test system respectively. For comparison purposes, sequential quadratic programming, genetic algorithm and key cutting algorithm were applied to solve the test system with various cases. Each method was challenged by solving given optimal power flow problems of 30 trials randomly. Minimum, average, maximum and standard deviation of the 30 trial solutions for all test system obtained by each method were evaluated and shown in table 19. Table 20 showed the comparison of CPU time spent by each approach. The optimal control variables obtained by each method and all test system were shown in tables 21 - 23.

Table 17: *Variable limits and cost coefficients used for the 6-bus test system with non- smooth fuel cost function*

Variable	Limit		cost coefficients				
	Min.	Max.	a	b	c	d	e
Mag. V1 to V2(p.u.)	1.0,1.1	1.1,1.15	-	-	-	-	-
Ang. V2 to V6(◦)	-30	30	-	-	-	-	-
T1 to T2 (p.u.)	0.90	1.10	-	-	-	-	-
QC1 to QC2(MVAR)	0,0	5,5.5	-	-	-	-	-
PG1(MW),QG1(MVAR)	10,-20	100,100	0.0016	2	150	50	0.063
PG2(MW),QG2(MVAR)	10,-20	100,100	0.01	2.5	25	40	0.098

The results show that, for the 6-bus system, the KCA can again obtain the lowest averaged minimum cost function (609.6 ฿/h, 635.5 ฿/h and 614.4 ฿/h for KCA, GA and SQP respectively). Due to a small number of control variables to be optimized, the SQP can be an appropriate choice of solving the optimal power flow problem even the fuel-cost function is non-smooth. The result of the 14-bus system, when comparing the averaged minimum costs obtained among those methods, the SQP is the best method (1749.5 ฿/h for the SQP, 1991.0 ฿/h for the GA and 1998.6 ฿/h for the KCA). However, with calculation time comparison, the KCA method can obtain the solution with the second calculation average time consumed (16.69 s) better than GA-based, while the fast method is SQP consumed average time at 15.46 s. The last test system is the 30-bus test system. The KCA is the best method (518.7 ฿/h for the KCA, 526.3 ฿/h for the GA and 766.1 ฿/h for the SQP). However, the KCA is the slowest method to find the solution. The solution convergences for each test case were shown in figure 19 - 21. Although the SQP is the fastest method, the KCA can find the most accurate solution with satisfactory CPU time consumed.

Table 18: *Variable limits and cost coefficients used for the 14-bus test system with non- smooth fuel cost function*

Variable	Limit		cost coefficients				
	Min.	Max.	a	b	c	d	e
Mag. V1 to V14(p.u.)	0.95	1.10	-	-	-	-	-
Ang. V2 to V6(◦)	-30	30	-	-	-	-	-
T1 to T2 (p.u.)	0.90	1.10	-	-	-	-	-
PG1(MW),QG1(MVAR)	50,-20	200,150	0	2.00	0.0200	300	0.20
PG2(MW),QG2(MVAR)	20,-20	80,60	0	1.75	0.0175	200	0.22
PG3(MW),QG3(MVAR)	15,-15	50,62.5	0	1.00	0.0625	150	0.42
PG6(MW),QG6(MVAR)	10,-15	35,48.7	0	3.25	0.0834	100	0.30
PG8(MW),QG8(MVAR)	10,-10	30,40	0	3.00	0.0250	200	0.35

Table 19: *Variable limits and cost coefficients used for the 30-bus test system with non- smooth fuel cost function*

Variable	Limit		cost coefficients				
	Min.	Max.	a	b	c	d	e
Mag. V1 to V14(p.u.)	0.95	1.10	-	-	-	-	-
Ang. V2 to V6(◦)	-30	30	-	-	-	-	-
T1 to T2 (p.u.)	0.90	1.10	-	-	-	-	-
QG1 to QG9 (MVAR)	0	5	-	-	-	-	-
PG1(MW), QG1(MVAR)	50,-20	200,150	0.00160	2.00	150	50	0.063
PG2(MW), QG2(MVAR)	20,-20	80,60	0.01000	2.50	25	40	0.098
PG5(MW), QG5(MVAR)	15,-15	50,44.7	0.06250	1.00	0	0	0
PG8(MW), QG8(MVAR)	10,-15	35,62.5	0.00834	3.25	0	0	0
PG11(MW), QG11(MVAR)	10,-10	30,40	0.02750	3.00	0	0	0
PG13(MW), QG13(MVAR)	12,-15	40,48.7	0.02500	3.00	0	0	0

Table 20: *OPF Results Obtained by using SQP, GA and KCA for all Test System with non-smooth fuel cost function*

Test System	Method	Optimal total generation costs (฿/h)			
		Minimum	Maximum	Average	Deviation
6 bus	SQP	542.6	675.9	614.4	37.29
	GA	585.7	755.9	635.5	35.3
	KCA	543.0	666.4	609.6	27.6
14 bus	SQP	1345.0	2177.2	1749.5	218.95
	GA	1632.6	2374.6	1991.0	191.2
	KCA	1674.8	2576.4	1998.6	195.9
30 bus	SQP	505.2	4248.8	766.1	734.1
	GA	490.9	572.8	526.3	17.9
	KCA	492.4	547.1	518.7	13.2

Table 21: *Computational Time to using SQP, GA and KCA for all Test System with non-smooth fuel cost function*

Test System	Method	Computational Time Consumed (s)			
		Minimum	Maximum	Average	Deviation
6 bus	SQP	0.89	6.15	2.40	1.26
	GA	4.86	7.28	6.01	0.59
	KCA	3.13	7.39	4.81	0.97
14 bus	SQP	6.91	37.78	15.46	8.31
	GA	16.89	25.33	19.88	1.78
	KCA	9.91	28.02	16.69	4.13
30 bus	SQP	71.57	154.48	104.8	24.58
	GA	749.2	752.1	46.2	0.79
	KCA	896.14	1646.9	1255.4	171.14

Table 22: *Optimal solution by each method for the 6-bus system with non-smooth fuel cost function*

variable	SQP	GA	KCA
Gen1 (MW),(MVAR) Gen2 (MW),(MVAR)	99.9,91.0 42.0,27.3	98.0,81.3 48.0,26.6	100.0,23.7 41.31,51.2
MagV1 (p.u.), AngV1 (◦)	1.0158,0	1.0792,0	1.0273,0
MagV2 (p.u.), AngV2 (◦)	1.1021, 1.77	1.1182, -17.57	1.1123,-5.24
QC1(MVAR) QC2(MVAR)	4.3650 2.2527	4.7939 4.0315	1.1437 1.3333
T1(T3-4) (p.u.) T2(T5-6) (p.u.)	1.1000 1.0979	0.9958 0.9814	1.0161 0.9753
PG total (MW)	142.0	146.1	141.3

Table 23: *Optimal solution by each method for the 14-bus system with non-smooth fuel cost function*

Variable	SQP	GA	KCA
Gen1 (MW),(MVAR)	81.4, 90.5	145.1, 121.5	144.1, 147.3
Gen2 (MW) (MVAR)	79.9, 4.9	76.3, 51.0	71.7, 39.2
Gen3 (MW),(MVAR)	44.4, 6.0	49.8, -4.5	45.8, 48.2
Gen6 (MW),(MVAR)	27.6, 16.8	17.2, 47.1	18.43, 13.4
Gen8 (MW),(MVAR)	29.9, 11.8	18.4,37.6	20.5, -9.0
MagV1 (p.u.),AngV1 (◦)	0.9936, 0	0.9805,0	1.0747,0
MagV2 (p.u.),AngV2 (◦)	0.9624,0.40	1.0279,0.37	1.0465,1.2941
MagV3 (p.u.),AngV3 (◦)	0.9518,-1.18	1.0893,12.87	1.0647,2.4706
MagV6 (p.u.),AngV6 (◦)	0.9984,-4.05	1.0280,-5.95	1.0129,14.4706
MagV8 (p.u.),AngV8 (◦)	0.9707,-15.88	1.0996, -10.05	1.0547,-3.6471
T1 (T4-7) (p.u.)	0.9003	0.9916	0.9220
T2 (T4-9) (p.u.)	0.9007	1.0598	0.9133
T3 (T5-6) (p.u.)	0.9016	1.0754	1.0514
T4 (T7-8) (p.u.)	0.9162	1.0065	0.9690
T5 (T7-9) (p.u.)	1.0981	1.0373	0.9282
PG total (MW)	263.4	307.0	300.7

Table 24: *Optimal solution by each method for the 30-bus system with non-smooth fuel cost function*

variable	SQP	GA	KCA
Gen1(MW),(MVAR)	187.0,179.6	192.2,141.1	187.0,198.2
Gen2 (MW) (MVAR)	34.1,40.1	29.6, 50.1	79.5,99.0
Gen5 (MW),(MVAR)	18.5,44.2	18.0, 76.6	15.0,71.0
Gen8 (MW),(MVAR)	23.8,59.6	28.5,44.0	19.5,46.4
Gen11 (MW),(MVAR)	26.3, 40.6	19.6, 43.4	28.0, 44.3
Gen13 (MW),(MVAR)	28.6,47.7	20.1,54.8	19.3,45.5
MagV1 (p.u.),AngV1(◦)	1.0429,0	1.0343,0	1.0929,0
MagV2 (p.u.),AngV2(◦)	1.0258, -4.53	1.0470, 0.5785	0.9529, 1.0265
Mag_5 (p.u.),AngV5 (◦)	1.0029, -2.21	1.0621, 1.3469	1.0182, 5.0588
MagV8 (p.u.),AngV8(◦)	0.9705, -8.13	0.9779, -3.6087	1.0200, 0.8235
MagV11 (p.u.),AngV11 (◦)	1.0080, 18.82	1.0689, -11.9386	1.0200, 7.4118
MagV13 (p.u.),AngV13(◦)	1.0107, -16.56	1.0219, -2.2383	1.0265, -7.1765
T1 (T6-9) (p.u.)	0.9345	1.0676	1.0420
T2 (T6-10) (p.u.)	0.8979	0.9987	0.9204
T3 (T4-12) (p.u.)	1.1004	0.9053	1.0082
T4(T25-27) (p.u.)	1.0229	1.0597	0.982
QC10(MVAR)	4.5406	4.2966	3.4902
QC12(MVAR)	4.7037	0.6063	0.4706
QC15(MVAR)	4.9648	3.9369	4.5686
QC17(MVAR)	3.6238	4.9800	3.6078
QC20(MVAR)	4.8405	0.0160	3.7451
QC21(MVAR)	0.7207	1.6031	1.5490
QC23(MVAR)	4.9635	4.8677	2.3137
QC24(MVAR)	3.9957	0.8845	4.4706
QC29(MVAR)	4.4840	0.0847	2.4314
PG total (MW)	318.5	308.2	348.4

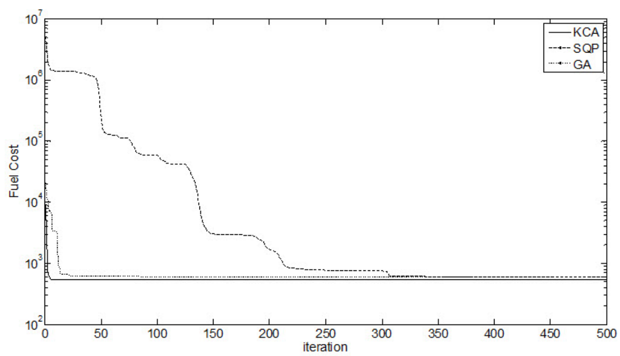

Fig.19: *Convergence characteristics of each method for the 6-bus test system with non-smooth fuel cost function*

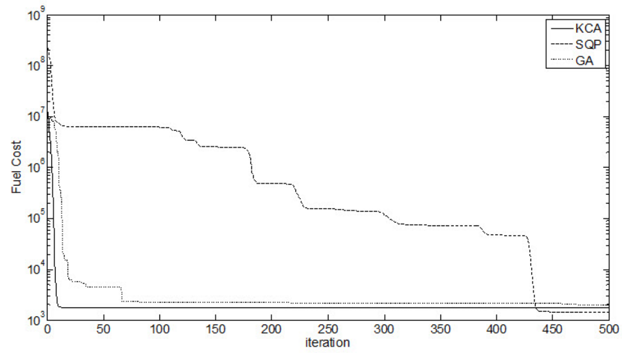

Fig.20: *Convergence characteristics of each method for the 14-bus test system with non- smooth fuel cost function*

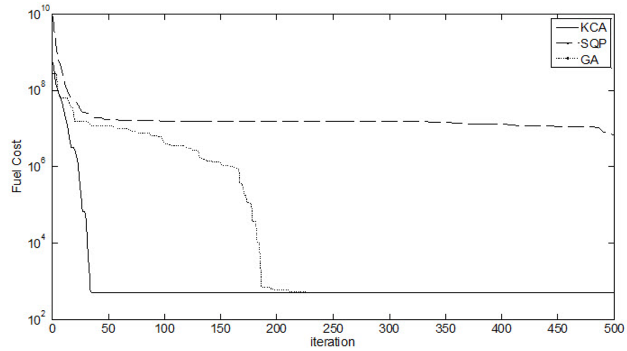

Fig.21: *Convergence characteristics of each method for the 30-bus test system with non- smooth fuel cost function*

5. CONCLUSIONS

This paper described the use of key cutting algorithm to find optimal power flow solutions. This work was conducted by 30 trials for both smooth and non-smooth fuel-cost functions in which four standard IEEE test power systems (6-bus, 14-bus and 30-bus) were employed. The test also applied the SQP, the genetic algorithm (GA) and particle swarm optimization of 30 trials each for comparison. The results showed that the key cutting algorithm can be the accurate method with satisfactory time consumed among all as it gives the smallest standard deviation of the 30 trial solutions for every test case. It also revealed that the SQP is suitable for optimal power flow problems with smooth quadratic fuel-cost functions where the total number of buses is small.

6. ACKNOWLEDGEMENT

This work supported by the Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program (Grant No. PHD/0047/2551) to Uthen Leeton and Thanatchai Kulworawanichpong are acknowledged.

References

[1] A.H. El-Abiad and F.J. Jaimes, “A method for optimum scheduling of power and voltage magnitude,” *IEEE Transactions. Power Apparatus and System.*, Vol. PAS-88, pp. 413 - 422, 1969.

[2] M.O. Mansour and T.M. Abdel-Rahman, ”Nonlinear VAR optimization using decomposition and coordination”, IEEE Transactions. Power Apparatus and System, Vol. PAS-103, pp. 246 - 255, 1984.

[3] H.W. Dommel and W.F. Tinney, ”Optimal power flow solutions”, IEEE Transactions. Power Apparatu. System, Vol. PAS-87, pp. 1866 - 1876, 1968.

[4] A.J. Wood and B.F. Wollenburg, Power Generation Operation and Control, 2 (John Wiley & Sons, 1996)

[5] Y. Hong-Tzer, Y. Pai-Chuan and H. Ching-Lien, ”Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions”, IEEE Transactions Power System, Vol. 11, pp. 112 - 118, 1996.

[6] W. Kit Po, ”Computational intelligence applications in unit commitment, economic dispatch and load flow”, The 4th International Conference on Advances in Power System Control, Operation and Management (APSCOM-97), vol. 1, pp. 54-59, 1997.

[7] D.C. Walters and G.B. Sheble, ”Genetic algorithm solution of economic dispatch with valve point loading”, IEEE Transactions on Power System, Vol. 8, pp. 1325-1332, 1993.

[8] D. Khanh and W. Kit Po, ”Artificial intelligence-based machine-learning system for thermal generator scheduling”, IEE Proceeding. Generation, Transmission and Distribution, Vol. 142, pp. 195 - 201, 1995.

[9] J.T. Ma and L.L. Lai, ”Evolutionary programming approach to reactive power planning”, IEE Proceeding. Generation, Transmission and Distribution, Vol. 143, pp. 365 - 370, 1996.

[10] T.L. Nguyen and Q.A. Nguyen, ”Application of artificial bee colony algorithm (ABC) for reconfiguring distribution network”, The 10th International Conference on Computer Modelling and Simulation (ICCMS-2011), Vol. 1, pp. 102-106, 2010.

[11] D. Karaboga, ”A new design method based on artificial bees colony algorithm for digital IRR filters”, Journal of the Franklin Institute, Vol. 346, pp.328 - 348, 2009.

[12] D. Karaboga and B. Busturk, ”A powerful and efficient algorithm for numerical function optimization: artificial bee colony optimization”, Journal of Global Optimization, Vol. 39, pp. 459 - 471, 2007.

[13] J. Qin, ”A New Optimization Algorithm and Its Application - Key Cutting Algorithm”, In 2009 IEEE International Conference on Grey Systems and Intelligent Services IEEE Press, New York, pp. 1537-1541, 2009.

[14] P. Dutta and A. K. Sinha, ”Voltage Stability Constrained Multi-objective Optimal Power Flow using Particle Swarm Optimization”, 1st International Conference on Industrial and Information Systems, pp. 161 - 166, 8-11 August 2006.

[15] J. Nocedal, and S.J. Wright, Numerical optimization, (Springer, 2000).

[16] S.G. Nash and A. Sofer, Linear and nonlinear programming, (McGraw-Hill, 1996).

[17] Z. Haibo, Z. Lizi and M. Fanling, ”Reactive power optimization based on genetic algorithm”, International Power Conference on Power System Technology, pp.1448 - 1453, 18-21 August 1998.

[18] K. Somsai, A. Oonsivilai, A. Srikaew, and T. Kulworawanichpong, ”Optimal PI controller design and simulation of a static var compensator using MATLAB’s SIMULINK”, The 7th WSEAS International Conference on POWER SYSTEMS, Beijing, China, pp. 30 - 35, September 2007.

[19] The MathWorks Inc., Genetic Algorithms and Direct Search TOOLBOX, CD-ROM Manual, 2004.

[20] J. Hazra1 and A. K. Sinha, ”A Study on Real and Reactive Power Optimization using Particle Swarm Optimization”, International Conference on Industrial and Information Systems, pp. 323 - 328, 9-11 August 2007.

[21] Z. Haibo, Z. Lizi and M. Fanling, Reactive power optimization based on genetic algorithm, International Power Conference on Power System Technology vol. 2, 18-21 August, pp. 1448 - 1453, 1998.

[22] S. He., Q.H. Wu, J.R. Saunders, Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. IEEE Transactions Evolutionary Computation 13, 973-990, 2009.

[23] P.N. Biskas, N.P. Ziogos, A. Tellidou, C.E. Zoumas, A.G. Bakirtzis and V. Petridis, ”Comparison of two metaheuristics with mathematical programming methods for the solution of OPF”, Generation, Transmission and Distribution, Vol. 153, pp. 16 - 24. , 2006.

[24] M.J. Rider, C.A. Castro, M.F. Bedrinana and A.V. Garcia, ”Towards a fast and robust interior point method for power system applications”, Generation, Transmission and Distribution, Vol. 151, pp. 575 - 581, 2004.

[25] K. Vaisakh, P. Praveena and S. Rama Mohana Rao, ”Solving optimal power flow problems using bacterial swarm optimization algorithm”, International Conference Advance in Power Sys-

tem Control,Operation and Management (APSCOM'09, pp. 1 - 7), 2009.

[26] X.P. Zhang and E.J. Handschin, "Advanced implementation of UPFC in a nonlinear interior-point OPF", Generation, Transmission and Distribution, Vol. 148, pp. 489 - 496, 2006.

Uthen Leeton is a Ph.D. Student of the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, THAILAND. He received B.Eng. with first-class honour in Electrical Engineering from Suranaree University of Technology, Thailand (2007). His fields of research interest include a power system analysis, optimization and artificial intelligent techniques. He has joined the school since June 2009 and is a research assistant and a member in Power System Research Unit, Suranaree University of Technology.

Thanatchai Kulworawanichpong is an associate professor of the School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, THAILAND. He received B.Eng. with first-class honour in Electrical Engineering from Suranaree University of Technology, Thailand (1997), M.Eng. in Electrical Engineering from Chulalongkorn University, Thailand (1999), and Ph.D. in Electronic and Electrical Engineering from the University of Birmingham, United Kingdom (2003). His fields of research interest include a broad range of power systems, power electronic, electrical drives and control, optimization and artificial intelligent techniques. He has joined the school since June 1998 and is currently a leader in Power System Research, Suranaree University of Technology, to supervise and co-supervise over 15 postgraduate students.