Volume 9, Number 12, December 2015 ISSN 1881-803X

ICIC International

ICIC Express Letters

An International Journal of Research and Surveys

Indexed by *Ei Compendex (Elsevier)*
Scopus (Elsevier)
INSPEC (IET)

Published by ICIC International
http://www.ijicic.org/icicel.htm

ICIC Express Letters
Volume **9**, Number **12**, December **2015**

ICIC International ©2015 ISSN 1881-803X
pp. 3415–3422

A KALMAN FILTER APPROACH FOR SOLVING UNIMODAL OPTIMIZATION PROBLEMS

Zuwairie Ibrahim[1], Nor Hidayati Abdul Aziz[1,2], Nor Azlina Ab. Aziz[2]
Saifudin Razali[1], Mohd Ibrahim Shapiai[3], Sophan Wahyudi Nawawi[4]
and Mohd Saberi Mohamad[5]

[1]Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang
Pahang 26600, Malaysia
{ zuwairie; saifudin }@ump.edu.my

[2]Faculty of Engineering and Technology
Multimedia University
Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
{ hidayati.aziz; azlina.aziz }@mmu.edu.my

[3]Department of Electronic Systems Engineering
Malaysia-Japan International Institute of Technology
Universiti Teknologi Malaysia
Kuala Lumpur 54100, Malaysia
md_ibrahim83@utm.my

[4]Faculty of Electrical Engineering
[5]Faculty of Computing
Universiti Teknologi Malaysia
Skudai, Johor 81310, Malaysia
sophan@fke.utm.my; saberi@utm.my

Received May 2015; accepted July 2015

Abstract. *In this paper, a new population-based metaheuristic optimization algorithm, named Simulated Kalman Filter (SKF) is introduced. This new algorithm is inspired by the estimation capability of the Kalman Filter. In principle, state estimation problem is regarded as an optimization problem, and each agent in SKF acts as a Kalman Filter. Every agent in the population finds solution to optimization problem using a standard Kalman Filter framework, which includes a simulated measurement process and a best-so-far solution as a reference. To evaluate the performance of the SKF algorithm in solving unimodal optimization problems, it is applied to unimodal benchmark functions of CEC 2014 for real-parameter single objective optimization problems. Statistical analysis is then carried out to rank SKF results to those obtained by other metaheuristic algorithms. The experimental results show that the proposed SKF algorithm is a promising approach in solving unimodal optimization problems and has a comparable performance to some well-known metaheuristic algorithms.*
Keywords: Optimization, Metaheuristics, Kalman, Unimodal

1. **Introduction.** Optimization is often required in solving engineering problems. Exact optimization methods normally fail to solve complex nonlinear and multimodal problems that exist in most real world applications in reasonable computational time. Thus, metaheuristic optimization methods are often sought to solve these kinds of problems. Metaheuristic algorithms are general algorithms that can be adapted into solving a wide range of optimization problems. A variant of metaheuristic algorithms is population-based. They rely on collection of agents to look for a near optimum solution within a reasonable computational effort.

Several population-based metaheuristic algorithms have been developed over the past 20 years. Most of them are inspired by nature. According to [1], existing algorithms can be divided into four categories based on their source of inspiration: bio-inspired swarm intelligence (SI) based, bio-inspired (non-SI) based, physics or chemistry based, and those that are not inspired by nature. Bio-inspired algorithms dominate the nature-inspired algorithms classification category. Among the famous and relatively new bio-inspired algorithms are Particle Swarm Algorithm [2], Bee Colony Optimisation [3], Cuckoo Search [4] and Firefly Algorithm [5]. All of them belong to Swarm Intelligence based algorithm. Besides that, there are famous algorithms belonging to physics inspired algorithm such as Gravitational Search Algorithm (GSA) [6] and Black Hole (BH) algorithm [7].

GSA is a well-known population-based metaheuristic algorithm inspired by physical phenomenon of Newtonian gravity and motion [6]. Black Hole is a more recent population-based metaheuristic algorithm inspired by the physical phenomenon of black hole [7]. While there are many metaheuristic algorithms being inspired by nature, some researchers do look away from nature for their source of inspiration. One example of non-nature inspired optimization algorithm is Heuristic Kalman Algorithm (HKA) by Toscano and Lyonnet in 2009 [8]. This population-based optimization algorithm is based on Kalman estimation method.

In this paper, a new metaheuristic optimization algorithm named Simulated Kalman Filter (SKF) is proposed to solve unimodal optimization problems. Similar to HKA, this new algorithm is inspired by the estimation capability of Kalman Filter. Therefore, it is expected that SKF has a very fast convergence rate as HKA. However, instead of relying on the properties of Gaussian distribution as in HKA, SKF simulates the measurement process as individual agent's update mechanism acting as feedback givers in estimating the optimum without being tied up to any type of distribution.

This paper is organized as follows. In Section 2, the proposed SKF algorithm is presented. Section 3 explains the experimental parameters in evaluating the performance of SKF, followed by experimental results and discussion in Section 4. Finally, Section 5 summarizes and concludes the paper.

2. Simulated Kalman Filter (SKF). Kalman Filter is a well-known state estimation method of a dynamic system that is excited by a stochastic process and measurement noise. Ever since its introduction by R. E. Kalman in 1960 [9], it has been extensively explored and used in many applications [10, 11, 12].

2.1. Principle of SKF algorithm. Figure 1 shows the principle of SKF algorithm. SKF makes an attempt to solve optimization problem by finding an estimate of the optimum.

In modelling the optimization problem as an estimation process of the optimum, static model of Discrete Kalman Filter is employed because the optimum solution to be estimated is not time dependent. Thus, the state vector can be reduced to a scalar. The state vector then contains only one variable that holds an agent's estimated position in the search space. These estimated states are used in the calculation of fitness based on an objective function.

Based on the capability of Kalman Filter in state estimation, each agent in SKF is able to improve its estimation of the optimum. In the proposed SKF algorithm, each agent acts as an individual Kalman Filter. Consider there are N agents and t indicates the iteration number, the estimated state of the ith agent at a time t, $\mathbf{X}_i(t)$, is defined as:

$$\mathbf{X}_i(t) = \left\{x_i^1(t), x_i^2(t), \ldots, x_i^d(t), \ldots, x_i^D(t)\right\} \text{ for } i = 1, 2, \ldots, N \tag{1}$$

where x_i^d represents the estimated state of the ith agent in the dth dimension and D is defined as the maximum number of dimension. In an iteration t, a number of agents are involved in the calculation of fitness, and agent with the best fitness, $\mathbf{X}_{best}(t)$, is identified. The SKF algorithm requires a simulated measurement process, which is led

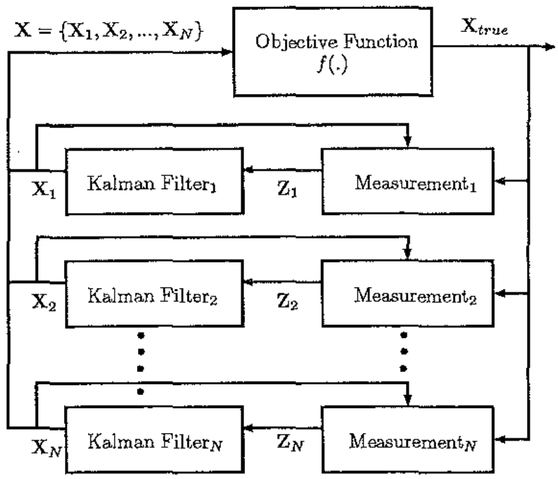

FIGURE 1. Principle of SKF algorithm

by a true value, $\mathbf{X}_{true}$. The $\mathbf{X}_{true}$ represents the best solution so-far, and will be updated when a better solution than $\mathbf{X}_{true}$ is found.

2.2. **SKF algorithm.** The proposed Simulated Kalman Filter (SKF) algorithm starts with initialization of the population. Then, the solutions of the initial population are evaluated and a true value is updated. Next, SKF algorithm iteratively improves its estimation by using the standard Kalman Filter framework which comprises predict, measure, and estimate. This process continues until the stopping condition is met.

2.2.1. *Initialization.* The SKF algorithm starts with random initialization of its agents' estimated state, $\mathbf{X}(0)$, within the search space. Besides the initial state estimate, the initial value of error covariance estimate, $P(0)$, the process noise, Q, and the measurement noise, R, are defined during initialization stage. Based on experiments, the value of $P(0)$, Q, and R are set to be 1000, 0.5, and 0.5, respectively, to give the best performance. The maximum number of iteration, $tMax$, is also initialized.

2.2.2. *Fitness evaluation, and $\mathbf{X}_{best}$ and $\mathbf{X}_{true}$ update.* The iteration begins with fitness calculation of the ith agent, $\mathrm{fit}_i(\mathbf{X}(t))$. Then, the $\mathbf{X}_{best}(t)$ is updated according to the type of problem. In minimization problem,

$$\mathbf{X}_{best}(t) = \min_{i\in 1,\ldots,N} \mathrm{fit}_i(\mathbf{X}(t)) \tag{2}$$

whereas, for maximization problem, Equation (3) is employed.

$$\mathbf{X}_{best}(t) = \max_{i\in 1,\ldots,N} \mathrm{fit}_i(\mathbf{X}(t)) \tag{3}$$

After that, the true value, $\mathbf{X}_{true}$, is updated. Note that the $\mathbf{X}_{true}$ represents the best solution so-far. Thus, $\mathbf{X}_{true}$ is updated if a better solution ($\mathbf{X}_{best}(t) < \mathbf{X}_{true}$ for minimization problem, or $\mathbf{X}_{best}(t) > \mathbf{X}_{true}$ for maximization problem) is found.

2.2.3. *Predict, measure, and estimate.* The search strategy follows three simple steps: predict-measure-estimate. Two sets of Kalman equations are adopted in SKF. The time-update equations are used to obtain the *a priori* estimates for the next time step. After the measurement process, estimation equations are used to obtain an improved *a posteriori* estimates.

In the prediction step, the following time-update equations:

$$\mathbf{X}(t|t-1) = \mathbf{X}(t-1) \tag{4}$$

$$P(t|t-1) = P(t-1) + Q \tag{5}$$

are used to make prediction of the state and error covariance estimates given the prior estimates. These estimates are called the *a priori* estimates. For static model of Discrete Kalman Filter, the state transition matrix takes the value of 1. And, since no control mechanism is employed in SKF, both the control input matrix and the control input vector are not used. This implies that there is no prediction in the state estimate because no control mechanism is used. However, the error covariance estimate, which is influenced by the process noise, is predicted.

The next step is measurement. Measurements act as feedback to estimation process. Measurement of each individual agent is simulated based on the following equation:

$$\mathbf{Z}_i(t) = \mathbf{X}_i(t|t-1) + \sin(rand \times 2\pi) \times |\mathbf{X}_i(t|t-1) - \mathbf{X}_{true}| \tag{6}$$

Given the predicted state estimate, $\mathbf{X}_i(t|t-1)$, measurement may take any random value from the predicted state estimate, $\mathbf{X}_i(t|t-1)$, to the true value, $\mathbf{X}_{true}$. A random element, $rand$, in $\sin(rand \times 2\pi)$ is the stochastic aspect of SKF algorithm. $rand$ takes a random value that is distributed uniformly in the range of $[0, 1]$. The probability density function (pdf) of the sine-wave distribution gives a high probability of occurrence near the extreme values, thus increasing the chance for more exploration. Since the difference between the predicted state estimate, $\mathbf{X}_i(t|t-1)$, and the true value, $\mathbf{X}_{true}$ is getting smaller as the iteration increases, exploration and exploitation can be compromised in SKF using Equation (6).

The final step is the estimation. During this step, Kalman gain, $K(t)$, is computed as follows:

$$K(t) = \frac{P(t|t-1)}{P(t|t-1) + R} \tag{7}$$

Then, the estimation equations, given by Equations (8) and (9), are used to improve the *a posteriori* estimates from the *a priori* estimates by making use of the measurement.

$$\mathbf{X}_i(t) = \mathbf{X}_i(t|t-1) + K(t) \times (\mathbf{Z}_i(t) - \mathbf{X}_i(t|t-1)) \tag{8}$$

$$P(t) = (1 - K(t)) \times P(t|t-1) \tag{9}$$

Using the measured position as feedback and influenced by the Kalman gain value, $K(t)$, each agent will give an estimate of the optimum position for that corresponding iteration. The next iteration is executed until the maximum number of iteration, $tMax$, is reached.

3. Experiments. The SKF algorithm is implemented using MATLAB. To evaluate the performance of the SKF algorithm in solving unimodal optimization problems, it is implemented to all three unimodal CEC 2014's benchmark functions [13]. There are three basic functions being considered in the unimodal benchmark functions. These basic functions are:

1. High Conditioned Elliptic Function

$$f_1(x) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} x_i^2 \tag{10}$$

2. Bent Cigar Function

$$f_2(x) = x_1^2 + \left(10^6\right) \sum_{i=2}^{D} x_i^2 \tag{11}$$

3. Discus Function

$$f_3(x) = \left(10^6\right) x_1^2 + \sum_{i=2}^{D} x_i^2 \tag{12}$$

TABLE 1. Summary of CEC 2014 unimodal benchmark functions

Function No.	Function	Ideal Fitness
1	$F_1(x) = f_1(\boldsymbol{M}(x - \boldsymbol{o}_1)) + F_1^*$	100
2	$F_2(x) = f_2(\boldsymbol{M}(x - \boldsymbol{o}_2)) + F_2^*$	200
3	$F_3(x) = f_3(\boldsymbol{M}(x - \boldsymbol{o}_3)) + F_3^*$	300

TABLE 2. SKF initialization parameters

Initialization Parameters	Values
Initial state estimate, $\mathbf{X}(0)$	$rand[-100, 100]$
Initial error covariance estimate, $P(0)$	1000
Process noise, Q	0.5
Measurement noise, R	0.5

TABLE 3. Experimental parameters

Experimental Parameters	Values
Number of agents, N	100
Number of iterations, $tMax$	2000
Number of dimensions, D	50
Number of runs, $runMax$	50

All the benchmark functions are minimization problems. They are shifted to global optimum, $\boldsymbol{o}_i$, and scalable. Different rotation matrix, $\boldsymbol{M}_i$, are assigned to each basic function. These benchmark functions are summarized in Table 1. The search space is the same for all the benchmark functions, which is $[-100, 100]$ for all dimensions. The MATLAB data files were downloaded from [14].

In SKF, 4 parameters need to be set during initialization stage. The initialization parameters are listed in Table 2.

In order to compare SKF results with those obtained by other metaheuristics, which are HKA, GSA and BH algorithm, all the other three algorithms were implemented in the same platform and subjected to the same parameter settings as listed in Table 3. The stopping criterion is set to be the maximum number of iteration for all algorithms. Boxplots are used to show the quality and consistency of the algorithms' performance. Since all the benchmark functions in the benchmark suite is a minimization problem, lower boxplot indicates a better quality solution. On the other hand, size of the boxplot represents its variance. Thus, smaller boxplot's size indicates a better consistency in the algorithm's performance.

To compare the results of all algorithms statistically, the mean fitness of the algorithms for each benchmark function is used. Friedman statistical test for non-parametric data with significance level $\alpha = 0.05$ is chosen for comparison purposes. This test is chosen because the nature of the solutions is not normally distributed. Non-parametric tests are encouraged to come out with the analysis of continuous optimization problems in multi-problem analysis [15]. In Friedman test, the algorithms are first ranked based on their mean fitness for each benchmark function. Then, the average rank for each algorithm is computed. These average Friedman rank is then used to calculate the Friedman statistical value given by Equation (13).

$$\chi^2 = \frac{12}{nk(k+1)} \sum (R_{avg} * n)^2 - 3n(k+1) \tag{13}$$

where n is number of algorithms, k is number of benchmark functions, and R_{avg} is the average Friedman rank. Friedman test will reject the NULL hypothesis if the statistical value is greater than the critical value ($\chi^2 > \chi^2_{crit}$). The NULL hypothesis set here is that the algorithms tested have equivalent performance to one another.

4. **Results and Discussion.** This section presented the results of the proposed SKF algorithm over the 3 unimodal benchmark functions available in the CEC 2014 Benchmark Suite. The performance of the SKF algorithm is then compared statistically to those obtained by other metaheuristic algorithms mentioned in the literature.

4.1. **SKF vs HKA, GSA and BH.** The boxplots are presented to show quality of results for CEC 2014 benchmark functions for all four algorithms. The outliers are excluded for better observation of the variance. The convergence curves are presented to show the convergence rate of the algorithms for all three unimodal benchmark functions. The convergence curves are plotted based on the mean fitness for each iteration until iteration 1000. The results of experiments when all the four algorithms are applied to the benchmark functions are presented as follows.

Figure 2 shows the boxplots for unimodal benchmark functions. It can be seen from Figure 2 that SKF gives the best and the most consistent performance for all unimodal functions. These are reflected by the position and size of SKF's boxplots for all three functions.

The convergence curves for Function 1 (Rotated High Conditioned *Elliptic* Function), Function 2 (Rotated *Bent Cigar* Function), and Function 3 (Rotated *Discus* Function) are shown in Figure 3. It can be seen from Figure 3 that the Kalman-based optimizers, which are SKF and HKA have a higher convergence rate compared to the other two

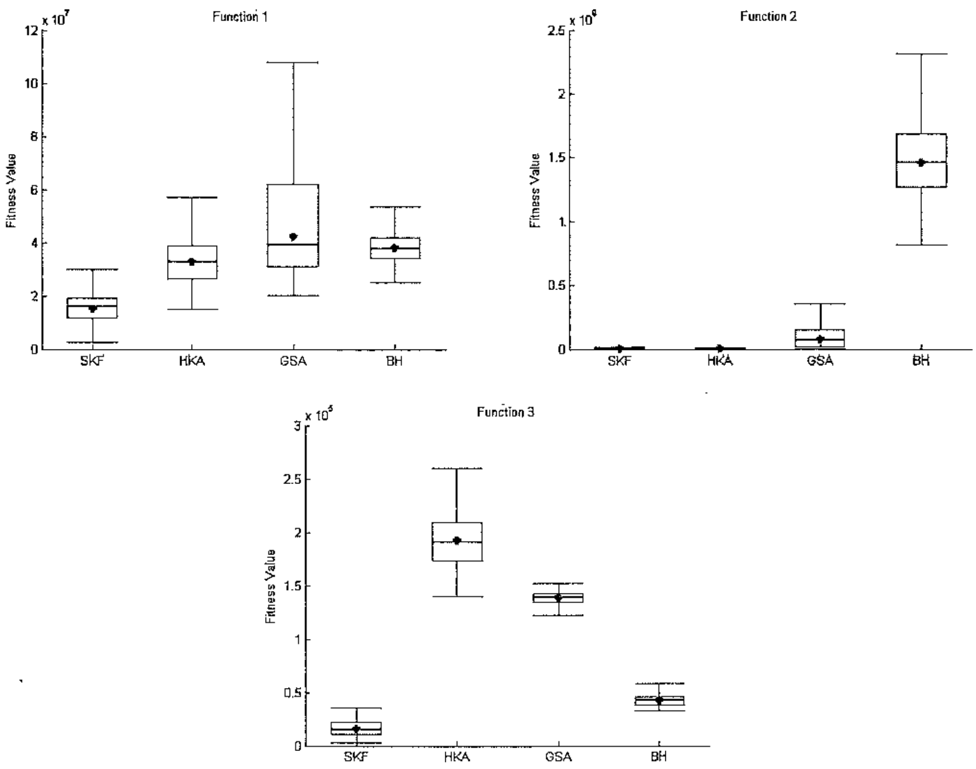

FIGURE 2. Results of experiments for unimodal benchmark functions

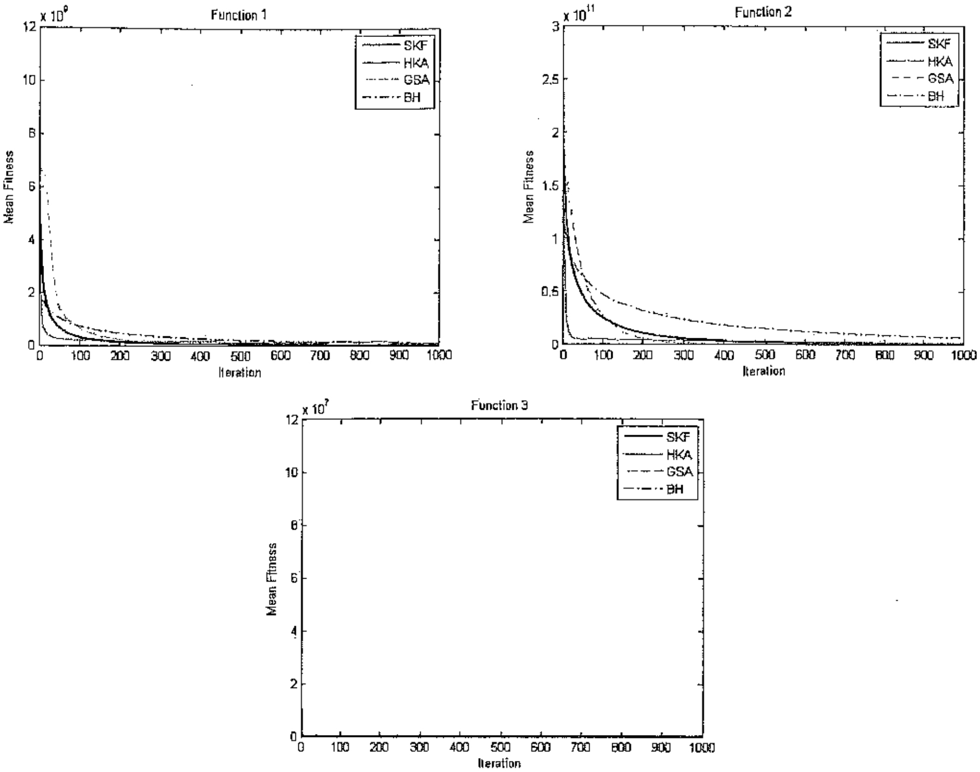

FIGURE 3. Convergence curve comparison for unimodal benchmark functions

algorithms, thus managed to locate the global minimum faster. This trend is considered unique compared to the other existing algorithms.

4.2. **Statistical analysis.** Statistical analysis is performed to rank the performance of SKF algorithm versus the other three metaheuristic algorithms. For this purpose, Friedman test is used to evaluate whether there is a significant difference between the algorithms' performance. The performance of the algorithms is ranked based on the mean value over the total number of runs for all three unimodal benchmark functions. The mean value calculated is inclusive of the outliers and is shown in the boxplot using the * symbol.

Table 4 shows the mean values for all the four algorithms for every benchmark function. The best solution for each benchmark function is marked in **bold**. Due to the fact that in Friedman test, the algorithms need to be ranked based on their mean fitness for each

TABLE 4. Mean value and Friedman rank of SKF, HKA, GSA, and BH

Function No.	SKF	HKA	GSA	BH
1	17370000	33716000	69128000	38451000
	1	2	4	3
2	18365000	122180	123250000	1481600000
	2	1	3	4
3	16118	192690	138080	43235
	1	4	3	2
Average Friedman Rank	1.33	2.33	3.33	3

benchmark function, the Friedman rank for all the four algorithms for every benchmark function is shown beneath the corresponding mean value in the same table. The average Friedman rank is then calculated for each algorithm. According to Friedman test, SKF is ranked the best among the four algorithms.

The Friedman statistical value is calculated using the average Friedman rank according to Equation (13) and is compared to the critical value according to chi-square distribution with 3 degrees of freedom. Statistically, Friedman test accepts the NULL hypothesis. Friedman test shows no significant difference exists between the algorithms since the statistical value ($\chi^2 = 4.2$) is less than the critical value ($\chi^2_{crit} = 7.815$).

5. **Conclusions.** In this paper, a new population-based metaheuristic optimization algorithm based on Kalman Filter approach, named Simulated Kalman Filter (SKF) is introduced to solve unimodal optimization problems. In evaluating our proposed algorithm, we have tested it using CEC 2014 unimodal benchmark functions and compared its performance to some existing metaheuristic algorithms. The results obtained show that SKF is able to converge to near-optimal solution and has a comparable performance to HKA, GSA and BH algorithm in solving unimodal optimization problems.

REFERENCES

[1] I. J. Fister, X. S. Yang, I. Fister, J. Brest and D. Fister, A brief review of nature-inspired algorithms for optimization, *Electrotechnical Review*, vol.8, no.3, 2013.

[2] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conf. on Neural Networks*, Perth, Australia, vol.IV, pp.1942-1948, 1995.

[3] D. Teodorovic and M. Dell' Orco, Bee colony optimization – A cooperative learning approach to complex transportation problems, *Abstracts of the 10th EWGT Meeting and the 16th Mini EURO Conference*, Poznan, 2005.

[4] X. S. Yang and S. Deb, Cuckoo search via Lévy flights, *Proc. of World Congress on Nature & Biologically Inspired Computing*, USA, pp.210-214, 2009.

[5] X. S. Yang, Firefly algorithm, stochastic test functions and design optimization, *Int. J. Bio-Inspired Computation*, vol.2, no.2, pp.78-84, 2010.

[6] E. Rashedi and H. Nezamabadi-pour, GSA: A gravitational search algorithm, *Inf. Sci. (Ny)*, vol.179, no.13, pp.2232-2248, 2009.

[7] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Inf. Sci.(Ny)*, vol.222, pp.176-184, 2013.

[8] R. Toscano and P. Lyonnet, Heuristic Kalman algorithm for solving optimization problems, *IEEE Trans. Systems, Man, and Cybernetics. Part B, Cybernetics*, vol.39, no.5, pp.1231-1244, 2009.

[9] R. E. Kalman, A new approach to linear filtering and prediction problems, *Trans. ASME-Journal Basic Eng.*, vol.82 (Series D), pp.35-45, 1960.

[10] F. Auger and M. Hilairet, Industrial applications of the Kalman filter: A review, *IEEE Trans. on Industrial Electronics*, vol.60, no.12, pp.5458-5471, 2013.

[11] Y. Mohammed and A. Alzubaidi, Missile position tracking using Kalman filter, *IOSR J. Eng.*, vol.04, no.02, pp.43-45, 2014.

[12] A. Toloei and S. Niazi, State estimation for target tracking problems with nonlinear Kalman filter algorithms, *Int. J. of Computer Applications*, vol.98, no.17, pp.30-36, 2014.

[13] J. Liang, B. Qu and P. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, *Technical Report 201311*, 2013.

[14] *Public Site. (n.d.)*, http://web.mysites.ntu.edu.sg/epnsugan/publicsite/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2Fepnsugan%2FPublicSite%2FShared%20Documents%2FCEC-2014&View=%7BDAF31868-97D8-4779-AE49-9CEC4DC3F310%7D, 2015.

[15] J. Derrac, S. Garcia, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.*, vol.1, no.1, pp.3-18, 2011.