Estimation-based Metaheuristics: A New Branch of Computational Intelligence

Nor Hidayati Abdul Aziz, Zuwairie Ibrahim, Saifudin Razali
Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang
26600 Pahang, Malaysia
pee14004@stdmail.ump.edu.my; {zuwairie; saifudin}@ump.edu.my

Nor Azlina Ab. Aziz
Faculty of Engineering and Technology
Multimedia University
75450 Melaka, Malaysia
azlina.aziz@mmu.edu.my

Abstract—In this paper, a new branch of computational intelligence named estimation-based metaheuristic is introduced. Metaheuristic algorithms can be classified based on their source of inspiration. Besides biology, physics and chemistry, state estimation algorithm also has become a source of inspiration for developing metaheuristic algorithms. Inspired by the estimation capability of Kalman Filter, Simulated Kalman Filter, SKF, uses a population of agents to make estimations of the optimum. Each agent in SKF acts as a Kalman Filter. By adapting the standard Kalman Filter framework, each individual agent finds an optimization solution by using a simulated measurement process that is guided by a best-so-far solution as a reference. Heuristic Kalman Algorithm (HKA) also is inspired by the Kalman Filter framework. HKA however, explicitly consider the optimization problem as a measurement process in generating the estimate of the optimum. In evaluating the performance of the estimation-based algorithms, it is implemented to 30 benchmark functions of the CEC 2014 benchmark suite. Statistical analysis is then carried out to rank the estimation-based algorithms' results to those obtained by other metaheuristic algorithms. The experimental results show that the estimation-based metaheuristic is a promising approach to solving global optimization problem and demonstrates a competitive performance to some well-known metaheuristic algorithms.

Keywords— metaheuristic optimization; estimation-based; Kalman Filter; SKF; HKA

1. Introduction

Optimization is often required in solving engineering problems. The growing complexity of real-world engineering problems has turn engineers to metaheuristic methods to save on computation time [1]. The development and enhancement of metaheuristic algorithms are still very active, even after the introduction of No Free Lunch (NFL) theorems by Wolpert and Macready in 1997 [2]. Accepting the fact that the NFL theorems suggested that there are no universally better algorithms, the research now is focusing on designing or finding the best algorithms for at least most types of problems. The search is still ongoing, and thus making this field of study an open problem.

Metaheuristic algorithms are general algorithms intended to solve global optimization problem in reasonable practical time. There are many ways of classifying metaheuristic algorithms. One way of classifying them is by looking at their source of inspiration [3]. The inspirations are typically related to nature. Genetic Algorithm [4] and Particle Swarm Optimization [5] for example, are classified as biological-inspired algorithms. Gravitational Search Algorithm [6] and Black Hole algorithm [7] on the other hand, are classified under physics/chemistry-inspired algorithms because they are inspired by the physical phenomenon of gravity and black hole respectively. Estimation-based method also has been a source of inspiration for some metaheuristic algorithms, thus can be considered as a new branch of metaheuristic algorithms classification. This type of algorithms mimics estimation algorithm in solving optimization problem.

A newly introduced estimation-based metaheuristic algorithm is Simulated Kalman Filter (SKF) [8]. This algorithm uses Kalman Filter [9], a state estimation algorithm as its source of inspiration. The SKF has been introduced initially to solve continuous, unimodal optimization problems. It has been recently tested for all benchmark functions of the CEC 2014 benchmark suite by Ibrahim et al. [10]. Ever since its introduction, many enhancements and variations of SKF has been introduced. Md. Yusof et al. [11-13] have extended the SKF algorithm into three different versions of SKF algorithms to deal with combinatorial optimization problems. The discrete type SKF algorithm is proven to be a promising approach to solving Airport Gate Allocation Problems (AGAP) [14] and peak detection problem of EEG [15]. The first attempt to improve the SKF performance is by hybridizing the SKF algorithm with PSO algorithm [16]. This new SKF-PSO hybrid is found to be superior to both SKF and PSO algorithm individually. Another estimation-based metaheuristic algorithm is Heuristic Kalman Algorithm [17]. It is introduced by Toscano and Lyonnet back in 2009. HKA has found to be useful as a simple and effective tuning strategy for PID controllers [18].

This paper is intended to study the effectiveness of estimation-based metaheuristic algorithms in solving global optimization problems of CEC 2014, which includes unimodal, simple multimodal, hybrid and composition problems. The rest of this paper is organized as follows: Section 2 describes the foundation of Kalman Filter, followed by a brief comparison between the SKF algorithm with the HKA. Section 3 explains the experimental parameters in evaluating the performance of estimation-based metaheuristic algorithms. The experimental results and discussion are presented and discussed in Section 4. Finally, Section 5 summarizes and concludes the paper.

2. ESTIMATION-BASED METAHEURISTIC ALGORITHMS

The Simulated Kalman Filter (SKF) and the Heuristic Kalman Algorithm (HKA) are two versions of estimation-based metaheuristic optimization algorithms inspired by the Kalman Filter estimation method. In this section, a brief description of Kalman Filter will be introduced followed by a brief comparison of SKF and HKA approaches in solving global optimization problems.

2.1. Discrete Kalman Filter

Discrete Kalman Filter is an efficient recursive filter, introduced by R. E. Kalman to estimate the state variable x, of a discrete time controlled process using a series of noisy measurements z. Let x_t, be the state vector containing the system's variable of interest at time t. In Kalman Filter model, x_t, evolves from the previous time step x_{t-1}, in accordance to (1).

$$x_t = A_t x_{t-1} + B_t u_t + w_t \tag{1}$$

where A_t is the state transition matrix, B_t is the control input matrix, u_t is the control input vector, and w_t is the process noise vector. At time t, the measurement vector z_t, can be represented as (2).

$$z_t = H_t x_t + v_t \tag{2}$$

where H_t is the measurement matrix, and v_t is the measurement noise vector.

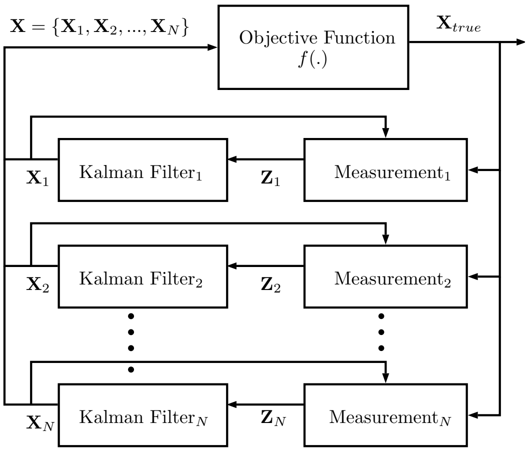
Discrete Kalman Filter, as a recursive estimator, needs only the estimated state from the previous time step $\hat{x}_{t-1}$, and current measurement z_t, to compute the estimate of the current state. At any given time, the state of the filter is represented by two variables, the state estimate $\hat{x}$, and the error covariance matrix P, as a measure of the estimated accuracy of the state estimate. Kalman Filter uses two sets of recursive equations. The first set of equations is called the time-update equations, whereas the second set of equations is called the measurement-update equations. Eq. (3) and (4) show the time-update equations that produce an estimate of the state at the current time step. The superscript (^{T}) represents the transpose operation.

$$\hat{x}_{t|t-1} = A_t \hat{x}_{t-1} + B_t u_t \tag{3}$$

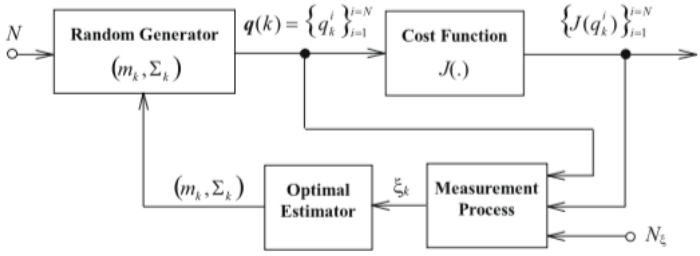
$$P_{t|t-1} = A_t P_{t-1} A_t^T + Q_t \tag{4}$$

where $\hat{x}_{t|t-1}$ is the predicted state estimate, and $P_{t|t-1}$ is the predicted error covariance matrix, P_{t-1} is the estimated error covariance matrix at previous time step, and Q_t is the process covariance matrix. The measurement-update equations are formulated as (5) and (6).

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t \left(z_t - H_t \hat{x}_{t|t-1} \right) \tag{5}$$



(a) SKF principle [8]



(b) HKA principle [11]

Figure 1: Principle comparison between SKF and HKA

$$P_t = \left(I - K_t H_t\right) P_{t|t-1} \tag{6}$$

where $\hat{x}_t$ is the estimated state vector at time t, P_t is the estimated error covariance matrix at time t, and K_t is called the Kalman gain. These measurement-update equations are employed to refine the state estimate by incorporating the current measurement into the predicted state estimate. The Kalman gain improves state estimate by minimizing the estimated error covariance during each iteration. In the Kalman Filter framework, the Kalman gain, K_t, is computed as in (7).

$$K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t\right)^{-1} \tag{7}$$

where R_t is the measurement covariance matrix.

2.2. Comparison of SKF with HKA

Basically, both SKf and HKA algorithms belong to a population based stochastic optimization group of algorithms since both rely on collection of agents in finding the estimate of the optimum solution. Despites the obvious similarity, SKF and HKA follow different approaches in finding the estimate. In SKF, each agent acts as a Kalman Filter, thus making individual estimation with the help from a simulated measurement process that is led by the best-so-far solution. HKA on the other hand, attempts to find the estimate of the optimum by considering the optimization problem as a measurement process by keeping on modifying the distribution, thus leading to one estimation only per iteration. While HKA embraces the assumption of Gaussian distribution, SKF works in a distributed free setting. Figure 1 compares between principle of SKF with the HKA's principle.

Consider a population of N agents with t indicates the iteration number, in SKF, the estimated state of the i^{th} agent at time t, $X_i(t)$ is given as (8).

$$X_i(t) = \left\{x_i^1(t), x_i^2(t), ..., x_i^d(t), ..., x_i^D(t)\right\} \tag{8}$$

where $x_i^d(t)$ represents the estimated state of the i^{th} agent in the d^{th} dimension with D is defined to be the maximum number of dimensions of the optimization problem. For HKA, for a collection of N vectors distributed around the mean vector, m_k, with a corresponding variance-covariance matrix, Σ_k with k indicates the iteration number, the i^{th} vector generated at iteration k is given by (9).

$$q_k^i = \left\{q_{1,k}^i ... q_{d,k}^i ... q_{n,k}^i\right\} \tag{9}$$

where $q_{d,k}^i$ represents the i^{th} vector in the d^{th} dimension with n is defined to be the maximum number of dimensions of the optimization problem. In both cases, (8) and (9) is the input of the objective or cost function for fitness calculation. Table 1 compares between the SKF algorithm with HKA.

Table 1: Pseudocode comparison between SKF and HKA.

Simulated Kalman Filter (SKF)	Heuristic Kalman Algorithm
1. Initialization (X(0), P(0), Q, R)	1. Initialization (N, N_ξ, α, m_0, Σ_0)
2. Fitness evaluation, and $X_{best}(t)$ and X_{true} updates	2. Generate sequence of N vectors using Gaussian random generator
3. Prediction for current time step	3. Measurement process
4. Simulated measurement	4. Optimal estimation
5. Estimate for the next time step	5. Initialization of the next time step
6. Check stopping condition	6. Check stopping condtion

During initialization, SKF randomly initialize the state estimate of a population of agents, X(0), over the search space in uniform distribution, and initialize the corresponding error covariance, P(0), the process noise, Q, and the measurement noise, R. HKA however, initialize the initial parameters of the Gaussian generator, which are the mean, m_0, and variance-covariance, Σ_0, for N number of points based on the lower bound and the upper bound the search space. Using that information, HKA generates a sequence of N vectors in normal distribution. Besides that, HKA needs to defne the number of best candidates, N_ξ, and also the slowdown coefficient, α.

Then, the fitness of each agent is evaluated. This information is used by SKF to update the $X_{best}(t)$ and X_{true} information to lead the search. The search in SKF consist of predict, measure and estimate step. The predict and estimate steps in SKF follow the standard Kalman Filter framework, with the measurement being simulated using (10).

$$Z_i(t) = X_i(t \mid t+1) + \sin\left(rand \times 2\pi\right) \times \left| X_i(t \mid t+1) - X_{true} \right| \tag{10}$$

where $X_i(t \mid t+1)$ represents the current state estimate of the prediction step and X_{true} is the best-so-far solution that is leading the search process. As the best so far solution, X_{true} will only be updated if $X_{best}(t)$ gives a better solution than X_{true} ($X_{best}(t) < X_{true}$ for minimization problem, or $X_{best}(t) > X_{true}$ for maximization problem). In minimization problem, $X_{best}(t)$ is given by (11), while for maximization problem, $X_{best}(t)$ is represented by (12).

$$X_{best}(t) = \min_{i \in 1,2,\ldots,N} fit\left(X_i(t)\right) \tag{11}$$

$$X_{best}(t) = \max_{i \in 1,2,\ldots,N} fit\left(X_i(t)\right) \tag{12}$$

In HKA, the fitness information is used in the measurement process where the average of the best candidates for k^{th} iteration is used as a measurement value given by (13).

$$\xi_k = \frac{1}{N_\xi} \sum_{i=1}^{N_\xi} q_k^i \tag{13}$$

Note that the best candidates in HKA are those which has the smallest cost function for minimization problem and the highest cost function for the maximization problem. The corresponding variance-covariance matrix, V_k, also being calculated by finding the average for the diagonal elements. The diagonal elements of V_k is represented by $vec^d(V_k)$ and is given by (14).

$$vec^d(V_k) = \frac{1}{N_\xi} \sum_{i=1}^{N_\xi} (q_{l,k}^i - \xi_{l,k})^2 \tag{14}$$

In the estimation of HKA, a slowdown coefficient, α, is used in the calculation of Kalman gain to avoid the estimator to converge too fast. The prior estimation before the measurement is then combined with the measurement, ξ_k, to come out with an optimal posterior estimation. At iteration k, the posterior estimated mean, m_k, and its variance-covariance matrix, Σ_k, are then used to initialize the random generator for the next time step $k+1$.

The stochastic element in SKF comes from the random initialization of the state estimate of the agents and also a uniformly distributed random value $rand$ in the simulated measurement equation. Equation (10) is used as the balancing mechanism between exploration and exploitation in SKF. In HKA however, the stochastic element only comes from the Gaussian random generator. Due to this, HKA can be seen as an evolutionary computation algorithm.

3. EXPERIMENTS

In order to test and validate the SKF algorithm, it is coded in MATLAB and implemented to all unimodal, simple multimodal functions, hybrid functions and composition functions of the CEC 2014 benchmark suite [19]. Note that all the benchmark functions in the CEC 2014 benchmark suite are minimization problems. The benchmark test functions in MATLAB are available online at: http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014.

Similar to [8] the performance of SKF and HKA for unimodal, simple multimodal, hybrid and composition functions is benchmarked against physics-based metaheuristics, which are Gravitational Search Algorithm (GSA), and Black Hole (BH) algorithm. All these algorithms were implemented in the same platform and subjected to the same initial and experimental settings. Specifically, for the SKF initial parameters, the initial error covariance, P(0), was set to 1000, and the process noise, Q, and the measurement noise, R, were set to 0.5. For HKA, the number of best candidates, N_ξ, is set to 10 and also the slowdown coefficient, α is set to 0.9 following the original setting of HKA. In terms of experimental parameters, all algorithms use 100 numbers of agents, and 2,000 maximum iterations, 50 dimensions, and 50 run times were selected.

For comparison purposes, the mean fitness and standard deviation of all the algorithms for each benchmark function were computed. Friedman and Holm statistical tests were then carried out to compare the performance of the SKF algorithm to the other metaheuristics. Friedman test was chosen rank the algorithms' performance overall benchmark functions while Holm test was responsible to test the 5% significant difference between the algorithms.

4. RESULTS AND DISCUSSION

In this section, the results of SKF performance over the 30 benchmark functions of CEC 2014 are presented and discussed. Table 2 shows the mean and standard deviation values for all the four algorithms for every benchmark function. The best solution for each benchmark function is marked in **bold**.

From Table 2, it can be seen that estimation-based metaheuristic algorithms which are SKF and HKA produce the best result for all 3 unimodal functions as presented in [8]. SKF and HKA also perform excellently at simple multimodal functions by leading all but 3 out of the 14 simple multimodal functions. Even for the 3 functions, they are lagging by just a few decimal points. Estimation-based metaheuristic algorithms also shown a competitive performance of hybrid and composition functions by leading 2 out of 6 in hybrid functions and 3 out of 8 in composition functions. They also produce a consistent performance looking at the small standard deviation value over the 50 runs for all function types, especially for HKA because the Gaussian spread of the estimated optimum tends to decrease over time.

Estimation-based metaheuristic algorithms also have a high convergence rate. This characteristic allows SKF and HKA algorithms to reach the global optimum faster than GSA and BH algorithms. The convergence curves for selected functions are shown in Fig. 2 for comparison.

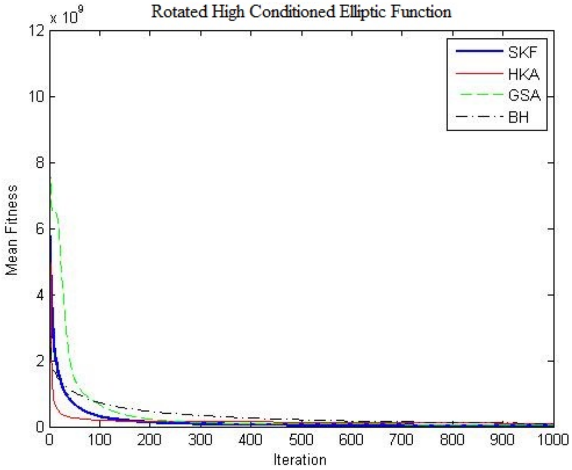
Statistically, Friedman statistic considering reduction performance distributed according to chi-square value of 15.93 with 3 degrees of freedom ranked SKF behind HKA, followed by GSA and then BH (see Table 3). Then, by using NxN post-hoc test at 5% significance level, Holm's procedure shows that SKF performance is on par to all algorithm as shown in Table 4. HKA on the other hand, outperform BH while performing on par with SKF and GSA. Note that for 5% significance level, Holm's procedure only rejects hypotheses that have an unadjusted p - value that is less than or equal to 0.01. These statistical results show that estimation-based metaheuristics is a very promising approach not only for unimodal optimization problems as in [10], but also for simple multimodal problems, hybrid and composition based problems.

5. CONCLUSION

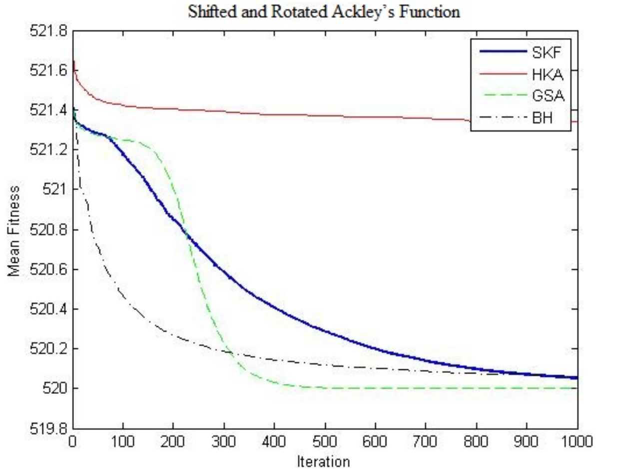
In this paper, a new classification of population-based metaheuristic optimization algorithms named estimation-based metaheuristic optimization is introduced. At the moment, there are already two algorithms parked under this new classification, the Simulated Kalman Filter (SKF) and the Heuristic Kalman Algorithm (HKA). Both algorithms takes inspiration from Kalman Filter, a well-known and effective state estimation algorithm. In evaluating the algorithms' performance under this proposed classification, the CEC 2014's benchmark suite has been used and the results are compared against physics-based algorithms, specifically the Gravitational Search Algorithm (GSA) and the Black Hole (BH) algorithms. Experimental results obtained show that SKF and HKA are able to converge to near optimal solution for all benchmark problems. Statistical analysis confirms that estimation-based metaheuristics is a promising approach and able to give a competitive performance compared to some well-known metaheuristic algorithms such as GSA and BH.

Table 2: Experimental results comparison of SKF and HKA with GSA and BH.

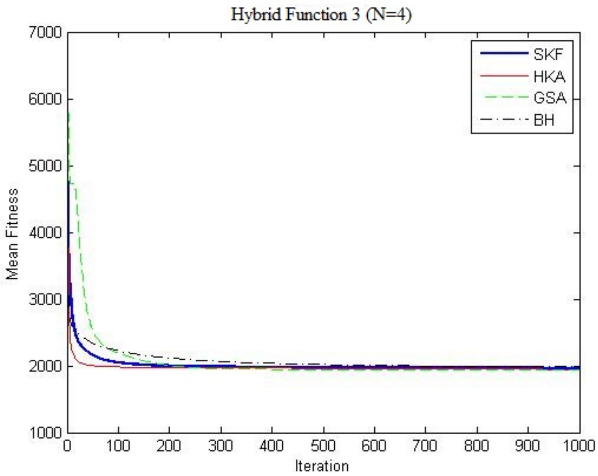
Function Type	Function Name		SKF	HKA	GSA	BH
Unimodal	Rotated High Conditioned Elliptic Function	Mean	**1.74E+07**	3.37E+07	6.91E+07	3.85E+07
		Std Dev.	8.10E+06	1.08E+07	1.05E+08	6.57E+06
	Rotated Bent Cigar Function	Mean	1.84E+07	**1.22E+05**	1.23E+08	1.48E+09
		Std Dev.	4.71E+07	1.31E+05	1.75E+08	2.98E+08
	Rotated Discus Function	Mean	**16118**	1.93E+05	1.38E+05	43235
		Std Dev.	7267.2	24687	7777.9	6400.9
Simple Multimodal	Shifted and Rotated Rosenbrock's Function	Mean	626.23	**448.04**	878.73	986.62
		Std Dev.	45.257	0.51159	179.07	78.542
	Shifted and Rotated Ackley's Function	Mean	520.01	520.28	**520**	520.04
		Std Dev.	0.012199	0.11817	9.80E-05	0.045258
	Shifted and Rotated Weierstrass Function	Mean	631.96	**601.62**	647.96	659.47
		Std Dev.	3.8817	0.30997	2.7796	5.3889
	Shifted and Rotated Griewank's Function	Mean	701.26	**700.16**	702.1	718.59
		Std Dev.	1.633	0.087132	1.715	3.2368
	Shifted Rastrigin's Function	Mean	**822.54**	823.71	1076.5	996.49
		Std Dev.	7.0145	0.087132	12.326	23.093
	Shifted and Rotated Rastrigin's Function	Mean	1059.6	**923.72**	1250.7	1224.9
		Std Dev.	30.575	5.0596	20.682	45.742
	Shifted Schwefel's Function	Mean	1426.2	**1399.3**	8193.2	5125
		Std Dev.	241.74	288.29	616.14	771.4
	Shifted and Rotated Schwefel's Function	Mean	6203.8	**1928.6**	9275.7	8614.5
		Std Dev.	893.56	383.71	654.93	978.07
	Shifted and Rotated Katsuura Function	Mean	1200.2	1200.1	**1200**	1200.8
		Std Dev.	893.56	383.71	0.0018287	0.2567
	Shifted and Rotated HappyCat Function	Mean	1300.6	**1300.2**	1300.5	1300.6
		Std Dev.	893.56	383.71	0.039138	0.04008
	Shifted and Rotated HGBat Function	Mean	1400.3	1400.4	**1400.3**	1400.3
		Std Dev.	893.56	383.71	0.022578	0.012858
	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Mean	1556.7	**1511.7**	1765.9	1898.6
		Std Dev.	893.56	1.9554	76.912	0.012858
	Shifted and Rotated Expanded Scaffer's F6 Function	Mean	1619.4	**1618.2**	1622.5	1621.9
		Std Dev.	0.83349	0.80653	0.27252	0.63537
Hybrid	Hybrid Function 1 (N=3)	Mean	2.82E+06	4.35E+06	**2.18E+06**	3.29E+06
		Std Dev.	1.47E+06	2.12E+06	9.80E+05	1.15E+06
	Hybrid Function 2 (N=3)	Mean	9.00E+06	34168	6.93E+07	**26431**
		Std Dev.	2.86E+07	36366	2.06E+08	47098
	Hybrid Function 3 (N=4)	Mean	1958.3	**1920.7**	1944	1969.1
		Std Dev.	29.325	1.0411	21.358	31.328
	Hybrid Function 4 (N=4)	Mean	35668	1.49E+05	59216	**25431**
		Std Dev.	17120	70721	21.358	8715.3
	Hybrid Function 5 (N=5)	Mean	3.11E+06	2.55E+06	**1.85E+06**	2.21E+06
		Std Dev.	1.85E+06	1.27E+06	4.32E+05	1.11E+06
	Hybrid Function 6 (N=5)	Mean	3473.4	**2351.3**	4133.9	3873.1
		Std Dev.	314.15	88.84	310.67	332.21
Composition	Composition Function 1 (N=5)	Mean	2649.3	2649.3	**2500**	2677
		Std Dev.	5.3598	1.712	1.02E-08	3.5418
	Composition Function 2 (N=3)	Mean	2666.5	2664.4	**2600.1**	2676.9
		Std Dev.	5.7677	0.25106	0.021272	9.6875
	Composition Function 3 (N=3)	Mean	2731.7	2706.8	**2700**	2748.2
		Std Dev.	3.6712	1.7408	1.86E-10	8.9883
	Composition Function 4 (N=5)	Mean	2792.9	**2727.8**	2800.1	2794.4
		Std Dev.	27.529	47.03	0.016801	23.965
	Composition Function 5 (N=5)	Mean	3905.2	**3053.7**	4789	4758.6
		Std Dev.	117.81	47.03	0.016801	161.39
	Composition Function 6 (N=5)	Mean	6934.6	**3199.4**	6083.9	11210
		Std Dev.	850.33	31.11	967.91	1145.9
	Composition Function 7 (N=3)	Mean	19573	3125.6	**3100.2**	18384
		Std Dev.	62411	31.11	0.014704	15797
	Composition Function 8 (N=3)	Mean	25821	3893.6	**3200**	2.05E+05
		Std Dev.	7629	31.11	0.0010457	38604



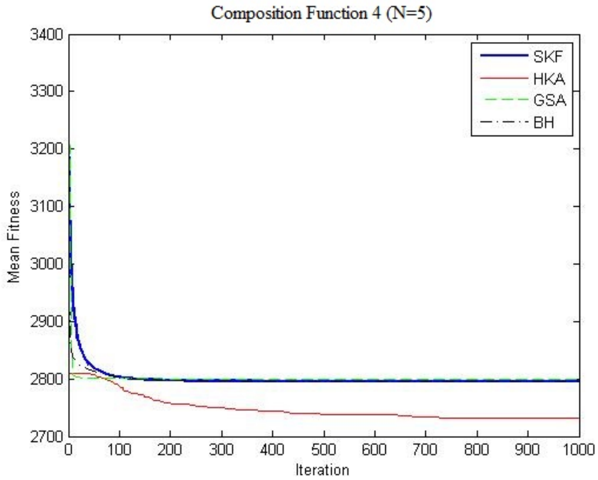
(a) Function 1 (Unimodal benchmark function)



(b) Function 5 (Simple multimodal benchmark function)



(c) Function 19 (Hybrid benchmark function)



(d) Function 26 (Composition benchmark function)

Figure 2: Selected convergence curves comparison.

Table 3: Friedman average ranking of the algorithms.

Algorithm	**Ranking**
SKF	2.3333
HKA	1.8333
GSA	2.6000
BH	3.1833

Table 4: Post-hoc Holm's analysis.

Algorithms	*z*	*p*	**Holm**
HKA vs BH	3.90	0.000096	0.008333
SKF vs BH	2.55	0.010772	0.01
HKA vs GSA	2.15	0.031555	0.0125
GSA vs BH	1.75	0.080118	0.016667
SKF vs HKA	1.35	0.177016	0.025
SKF vs GSA	0.80	0.423711	0.05

ACKNOWLEDGMENT

This work was supported by Universiti Malaysia Pahang PGRS under Project ID. GRS1503120, and Ministry of Higher Education FRGS under Project ID. FRGS/1/2015/TK04/MMU/03/2.

REFERENCES

[1] E.G. Talbi, Metaheuristics: from design to implementation. Wiley, 2009.

[2] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, 1997, pp. 67–82.

[3] 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.

[4] J. H. Holland, “Adaptation in Natural and Artificial Systems,” Ann Arbor: University of Michigan Press, 1975.

[5] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in Proc. of IEEE Int. Conf. on Neural Network, Perth, Australia, 1995, pp.1972–1978.

[6] E. Rashedi and H. Nezambadi-pour, “GSA: gravitational search algorithm,” Information Sciences, vol. 179, no. 13, 2009, pp. 2232–2248.

[7] A. Hatamlou, “Black Hole: A new heuristic optimization approach for data clustering,” Information Sciences, vol. 222, 2013, pp. 176–184.

[8] Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, M. I. Shapiai, S. W. Nawawi, and M. S. Mohamad, “A Kalman Filter approach for solving unimodal optimization problems,” ICIC Express Letters, vol. 9, no. 12, 2015, pp. 3415–3422.

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

[10] Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, and M. S. Mohamad, “Simulated Kalman Filter: A novel estimation-based metaheuristic optimization algorithm”, Advanced Science Letters, 2016, in press.

[11] Z. Md Yusof, Z. Ibrahim, I. Ibrahim, K. Z. Mohd Azmi, N. A. Ab. Aziz, N. H. Abdul Aziz, and M. S. Mohamed, “Angle Modulated Simulated Kalman Filter algorithm for combinatorial optimization problems,” ARPN Journal of Engineering and Applied Sciences, vol. 11, no. 7, 2015, pp.4854-4859.

[12] Z. Md Yusof, Z. Ibrahim, I. Ibrahim, K. Z. Mohd Azmi, N. A. Ab. Aziz, N. H. Abdul Aziz, and M. S. Mohamed, “Distance Evaluated Simulated Kalman Filter algorithm for combinatorial optimization problems,” ARPN Journal of Engineering and Applied Sciences, vol. 11, no. 7, 2015, pp.4904-4910.

[13] Z. Md Yusof, I. Ibrahim, S. Z. Satiman, Z. Ibrahim, N. H. Abdul Aziz, and N. A. Ab. Aziz, “BSKF: Binary Simulated Kalman Filter,” Third International Conference on Artificial Intelligence, Modelling and Simulation, 2015, pp.77-81.

[14] Z. Md Yusof, S. Z. Satiman, K. Z. Mohd Azmi, B. Muhammad, S. Razali, Z. Ibrahim, Z. Aspar, and S. Ismail, “Solving airport gate allocation problem using Simulated Kalman Filter,” In: International Conference on Knowledge Transfer 2015, 1-3 December 2015, Putrajaya Marriot Hotel, Malaysia.

[15] A. Adam, Z. Ibrahim, N. Mokhtar, M. I. Shapiai, M. Mubin, I. Saad, “Feature selection using angle modulated simulated Kalman filter for peak classification of EEG signals,” SpringerPlus, Vol. 5, No. 1580, 2016.

[16] B. Muhammad, Z. Ibrahim, K. H. Ghazali, K. Z. Mohd Azmi, N. A. Ab. Aziz, N. H. Abdul Aziz, and M. S. Mohamed, “A new hybrid Simulated Kalman Filter and Particle Swarm Optimization for continuous numerical optimization problems,” ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 22, 2015, pp.17171-17176.

[17] R. Toscano and P. Lyonnet, “Heuristic Kalman Algorithm for solving optimization problems,” IEEE Trans. on Systems, Man, and Cybernetics. Part B, Cybernetics, vol. 39, no. 5, 2009, pp. 1231–1244.

[18] R. Toscano and P. Lyonnet, “Robust PID controller tuning based on the Heuristic Kalman Algorithm,” Automatica, vol. 45, no. 9, 2009, pp. 2099-2106.

[19] J. J. Liang, B. Y. Qu, P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization,” Tech. Rep. 201311, Computational Intelligence Laboratory, Zhengzhou University, 2013.