International Journal of Engineering & Technology, 7 (4.27) (2018) 44-49

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Asynchronous Simulated Kalman Filter Optimization Algorithm

Nor Azlina Ab. Aziz[1], Zuwairie Ibrahim[2*], Nor Hidayati Abdul Aziz[3], Tasiransurini Ab. Rahman[4]

[1 3]Faculty of Engineering and Technology, Multimedia University, 75450 Bukit Beruang, Melaka, Malaysia.
[2]Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia.
[4]Faculty of Electrical and Electronic, Universiti Tun Hussein Onn Malaysia, 86400 Johor, Malaysia
**Corresponding author E-mail: zuwairie@ump.edu.my*

Abstract

Simulated Kalman filter (SKF) is an optimization algorithm which is inspired by Kalman filtering method. SKF was introduced as synchronous population-based algorithm. This work introduced a new variation of SKF which is SKF with asynchronous update mechanism, asynchronous-SKF (ASKF). In contrast to the synchronous implementation where the whole population go through each optimization step as a group, in ASKF an agent starts its optimization steps only after its preceding agent has completed all optimization steps. The performance of ASKF is compared against SKF using CEC2014 benchmark functions, where the ASKF is found to perform significantly better than the original SKF.

Keywords: *Asynchronous; Simulated Kalman Filter; Optimization.*

1. Introduction

A metaheuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space so that a near-optimal solution can be obtained [1]. In 2015 a new metaheuristic algorithm, SKF, had been proposed for continuous unimodal optimization problems [2]. It was introduced as population-based metaheuristics, where the search for optimal solution is conducted by a group of agents. The agents of SKF work like Kalman filters, where they go through prediction, measurement, and estimation process in every iteration. The measurement in SKF is a simulated measurement which is obtained using mathematical equation.
Many works had been conducted on SKF, where it had been modified for binary optimization problems [3] and combinatorial optimization problems [4-7]. Hybridization of SKF with particle swarm optimization (PSO) [8-9] and gravitational search algorithm (GSA) [10-11] had also been proposed with better performance reported. A parameterless SKF algorithm is proposed in [11]. SKF has also been applied for real world problems like, the adaptive beamforming in wireless cellular communication [12-13], airport gate allocation problem [14-15], feature selection of EEG signal [16], ARX system identification [17] and PCB drill path optimization [18].
As a population-based metaheuristic algorithm, the SKF's agents conduct the search for optimal solution through information sharing. The evaluation of the candidate solutions found by SKF agents and the Kalman filter's procedure of predict, measure and estimate are done iteratively. How the agents move from evaluation to the Kalman procedure, either as a group or individually is determined by the iteration strategy. The group-oriented iteration strategy is known as synchronous update while the individual-oriented iteration strategy is known as asynchronous update. So far, studies on SKF have been carried out based on synchronous update implementation, where every agent of the population need to complete the evaluation phase before the Kalman phase can begin. In this work, an asynchronous SKF (ASKF) is introduced. An agent in ASKF is evaluated and the agent completes all three Kalman phases, which are predict, measure, and estimate, before another agent begins its evaluation, prediction, measurement, and estimation. An iteration of ASKF completes after the whole population has completed these phases. The performance of ASKF is compared with the original SKF using CEC2014 benchmark function, where it is found that statistically ASKF is better than the original SKF.

2. The Original SKF Algorithm

The SKF algorithm follows the pseudocode shown in Figure 1. It starts with random initialization of the agents' estimated values, $\boldsymbol{X}_i(t)$. The estimated values represent candidate solutions of the problem to be solved.
In each iteration the fitness of the agents' estimate is evaluated using the problem to be solved such as the CEC2014's benchmark problems. Once the evaluation is completed, the agent with the best fitness value is identified as the best solution of the current population, $\boldsymbol{X}_{best}(t)$. Next, the best $\boldsymbol{X}_{best}(t)$ from the first iteration is selected as $\boldsymbol{X}_{true}$.
The agents are then updated following the Kalman filter procedure of predict, measure and estimate. During the prediction phase, the current predicted state, $\boldsymbol{X}_i(t|t+1)$, is assumed to be the estimated value;

$$X_i(t|t+1) = X_i(t) \tag{1}$$

The error covariant is also updated as follows;

$$P(t|t+1) = P_i(t) + Q \tag{2}$$

where $\boldsymbol{P}_i(t)$ and $\boldsymbol{P}(t|t+1)$ denote the current error covariant estimate and current transition error covariant estimate, respectively. Note that the error covariant estimate is influenced by the process noise, Q.
In SKF, measurements are simulated using an agent's prediction and $\boldsymbol{X}_{true}$. The dimensional wise calculation of measured value for dimension j^{th} of agent i^{th} is calculated as follows;

$$Z_i^j(t) = X_i^j(t|t+1) + \sin\left(2\pi r_i^j(t)\right) \times |X_i(t|t+1) - X_{true}| \quad (3)$$

The $r_i^j(t)$ is a random value within the range of [0,1]. The estimation phase follows the measurement phase and the estimated next value is updated using (4);

$$x_i^d(t+1) = x_i^d(t|t+1) + K(t) \times \left(z_i^d(t) - x_i^d(t|t+1)\right) \quad (4)$$

where $K(t)$ is the Kalman gain, which is calculated as follows;

$$K(t) = P(t|t+1)/(P(t|t+1) + R) \quad (5)$$

In (5), R is the measurement noise, which is suggested to be set to 0.5. Then, the current error covariant estimate is updated in estimation phase using (6);

$$P(t+1) = (1 - K(t)) \times P(t|t+1) \quad (6)$$

These steps continue until the maximum iteration is reached.

3. The Proposed Asynchronous SKF (ASKF)

The pseudocode of the proposed ASKF is presented in Figure 2. Similar to the original SKF, ASKF starts with random initialization of the population according to the problem's search space. However, the steps within the iteration are individually executed for ASKF. Therefore, in an iteration of ASKF, as soon as an agent is evaluated, its fitness is compared with $\boldsymbol{X}_{true}$. If the agent has found a better solution, then the $\boldsymbol{X}_{true}$ is immediately updated according to the estimated value of the agent. Thus, in ASKF, $\boldsymbol{X}_{best}(t)$ is not needed.
After the $\boldsymbol{X}_{true}$ comparison, the agent's state is immediately predicted. This is followed by the agent's measurement and state estimation. The prediction, measurement and estimation are carried using the same set of equations like the original SKF. When an agent completed its Kalman filter's procedures, next agent is selected to go through the same steps.

Table 1: The CEC2014 benchmark test suite (source: [17])

Types	No.	Functions	Ideal Fitness
Unimodal functions	f1	Rotated High Conditioned Elliptic function	100
	f2	Rotated Bent Cigar function	200
	f3	Rotated Discus function	300
Simple multimodal functions	f4	Shifted and Rotated Rosenbrock's function	400
	f5	Shifted and Rotated Ackley's function	500
	f6	Shifted and Rotated Weierstrass function	600
	f7	Shifted and Rotated Griewank's function	700
	f8	Shifted Rastrigin's function	800
	f9	Shifted and Rotated Rastrigin's function	900
	f10	Shifted Schwefel's function	1000
	f11	Shifted and Rotated Schwefel's function	1100
	f12	Shifted and Rotated Katsura function	1200
	f13	Shifted and Rotated HappyCat function	1300
	f14	Shifted and Rotated HGBat function	1400
	f15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's function	1500
	f16	Shifted and Rotated Expanded Scaffer's F6 function	1600
Hybrid functions	f17	Hybrid function 1 (N=3)	1700
	f18	Hybrid function 2 (N=3)	1800
	f19	Hybrid function 3 (N=4)	1900
	f20	Hybrid function 4 (N=4)	2000
	f21	Hybrid function 5 (N=5)	2100
	f22	Hybrid function 6 (N=5)	2200
Composition functions	f23	Composition function 1 (N=5)	2300
	f24	Composition function 2 (N=3)	2400
	f25	Composition function 3 (N=3)	2500
	f26	Composition function 4 (N=5)	2600
	f27	Composition function 5 (N=5)	2700
	f28	Composition function 6 (N=5)	2800
	f29	Composition function 7 (N=3)	2900
	f30	Composition function 8 (N=3)	3000

4. Experiment, Results & Discussion

The performance of the proposed ASKF is compared with the original SKF using CEC2014 Benchmark Test Suite for single-objective optimization. The test suite comprises of 30 functions consisting mixture of; three unimodal test suite, 13 simple multimodal test suite, six hybrid test suite, and eight composition test suites. The test functions are tabulated in Table 1.
The comparison is conducted using population of 100 agents, dimension size of 30, and maximum iteration of 3000. Each of the experiment is run 30 times.

```
Initialization of agents
Do{
    For every agents
        Evaluate fitness
    End for
    Identify X_best(t)
    Update X_true
    For every agent
        Predict
        Measure
        Estimate
    End for
}While not maximum iteration
```

Fig. 1: The original simulated Kalman filter algorithm.

```
Initialization of agents
Do{
    For every agents
        Evaluate fitness
        Update X_true
        Predict
        Measure
        Estimate
    End for
}While not maximum iteration
```

Fig. 2: The asynchronous simulated Kalman filter algorithm.

The averaged error value of the solution obtained by the algorithm with the ideal solution for each benchmark function is tabulated in Table 2. Better values are written in bold. The ASKF able to find better performance for 23 functions from the 30 test functions.
The boxplots for ASKF and the original SKF (labelled as S-SKF) are presented in Figures 3-6. The boxplots of ASKF are at lower position than the original SKF. The original SKF's boxplots also have bigger distribution than ASKF's. These boxplots illustrate the better consistency in the solutions' quality found by ASKF compared to the original SKF. The original SKF also produced more outliers in unimodal, hybrid, and composite functions. There are no outliers for both SKF and ASKF for the case of simple multimodal functions
The Wilcoxon signed rank test with significance level, α = 0.05 is chosen to provide an unbiased observation. The test gives statistical value of 122 which is smaller than 137, thus the null hypothesis of equivalent performance is rejected and ASKF is concluded to be significantly better than the original SKF.

The convergence of ASKF and the original SKF (labelled as S-SKF) are presented in Figures 7-10. For both algorithms, the fitness error rate decreased exponentially, but the original SKF's fitness error decreased more rapidly than ASKF's. In several functions, namely f6, f9, f11, f12, f16, f25, and f28, the ASKF achieved a lower error value.

5. Conclusion

A SKF algorithm that operates asynchronously is proposed in this work. In an iteration the agents of ASKF algorithm perform fitness evaluation and the Kalman procedure one after another. This is different than the original SKF where these steps are done simultaneously. Based on the experiment conducted using the CEC2014's benchmark suite, ASKF is found to perform significantly better than the original SKF. This finding shows the potential of ASKF as an efficient optimizer.

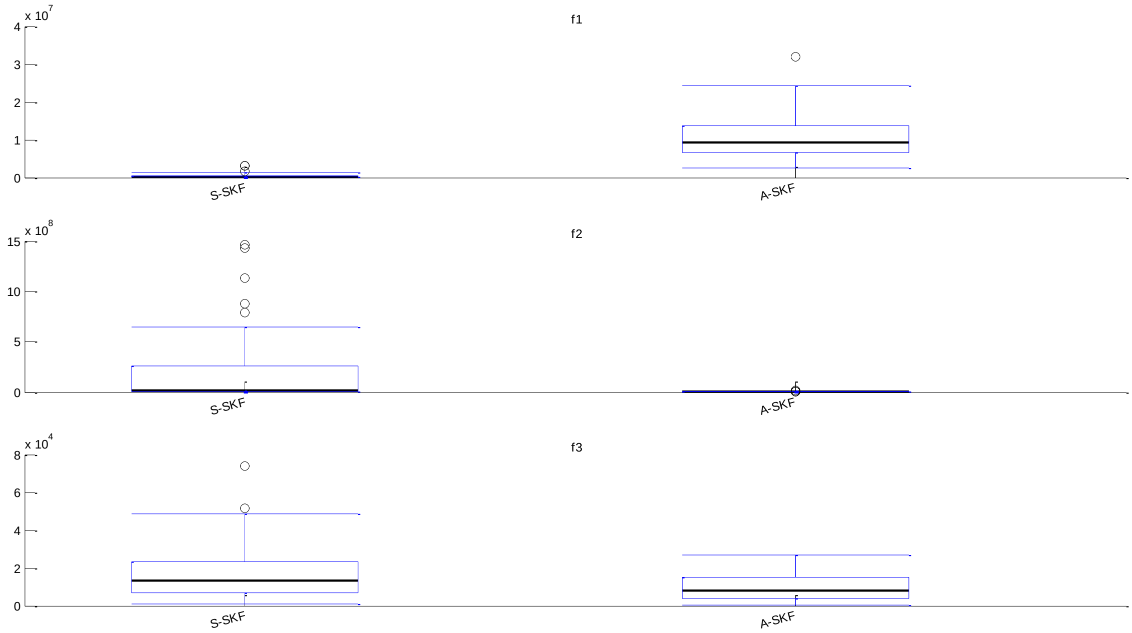

Fig. 3: Fitness error distribution of unimodal functions.

Table 2: Average error

Functions	Average Error (SKF)	Average Error (ASKF)
f1	**486000**	11000000
f2	2.45E+08	**1290000**
f3	18410	**9901**
f4	**36.46**	117.7
f5	20.02	**20.01**
f6	21.95	**18.17**
f7	0.1635	**0.08444**
f8	5.878	**5.473**
f9	90.87	**75.26**
f10	226.3	**162**
f11	2640	**2585**
f12	0.3592	**0.2099**
f13	0.4443	**0.3567**
f14	0.2593	**0.2273**
f15	21.92	**16.4**
f16	**10.6**	10.67
f17	**105000**	1170000
f18	11500000	**8560000**
f19	20.5	**19.85**
f20	29840	**24150**
f21	**261000**	555000
f22	621.7	**497.3**
f23	318.1	**316.1**
f24	231	**229.2**
f25	215.1	**214.3**
f26	**120.4**	**120.4**
f27	598.5	**547.6**
f28	**1574**	1610
f29	2477	**1189**
f30	5438	**3848**

The number of applications for optimization algorithm such as the ASKF is huge. For example, in computational science, SKF could optimizes the advanced system hardware, software, networking, and data management components needed to solve computationally demanding problems. In control engineering, tuning of PID controller parameters for an optimized control performance is a multi-objective optimization problem. The problem becomes particularly difficult if the plant to be controlled is an unstable, nonlinear and under actuated plant. Thus, ASKF could be used as multi-objective optimization tools for tuning of PID controller parameters. In economics, most of the optimization problems in economics are problems of constrained optimization: maximizing or minimizing some objective function subject to one or more constraints. One example is profit maximization problem for the competitive firm. Constraints handling can be integrated in ASKF and these economic problems can be solved using ASKF. In chemistry, ASKF could be used to find the global minimum, that is the lowest value of Potential Energy Surface (PES) in an N-atomic molecule. In mathematics, gradient-based approach could be replaced with population-based approach of ASKF for solving constrained, non-linear optimization problems.

Acknowledgement

This research is financially supported by the Malaysian Ministry of Education through the Fundamental Research Grant Scheme that is awarded to Universiti Malaysia Pahang (UMP) - RDU170106.

References

[1] Osman IH & Laporte G (1996), Metaheuristics : A Bibliography, *Ann. Oper. Res.*, Vol. 63, No. 5, pp. 513–628.

[2] Ibrahim Z, Abdul Aziz NH, Ab. Aziz NA, Razali S, Shapiai MI, Nawawi SW & Mohamad MS (2015), A Kalman Filter Approach for Solving Unimodal Optimization Problems, *ICIC Express Letters*, Vol. 9, Issue 12, pp. 3415-3422.

[3] Md Yusof Z, Ibrahim I, Satiman SN, Ibrahim Z, Abd Aziz NH, & Ab Aziz NA, BSKF: Binary Simulated Kalman Filter, *Third International Conference on Artificial Intelligence, Modelling and Simulation*, (2015), pp. 77-81.

[4] Md Yusof Z, Ibrahim I, Ibrahim Z, Abas KH, Ab Aziz NA, Abd Aziz NH & Mohamad MS, Local Optimum Distance Evaluated Simulated Kalman Filter for Combinatorial Optimization Problems, *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 892-901.

[5] Md Yusof Z, Ibrahim Z, Ibrahim I, Mohd Azmi KZ, Ab. Aziz NA, Abd Aziz NH & Mohamad MS (2016), Angle Modulated Simulated Kalman Filter Algorithm for Combinatorial Optimization Problems, *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, No. 7, pp. 4854-4859.

[6] Md Yusof Z, Ibrahim I, Ibrahim Z, Abas KH, Sudin S, Ab. Aziz NA, Abdul Aziz NH & Mohamad MS, Three Approaches to Solve Combinatorial Optimization Problems using Simulated Kalman Filter, *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 951–960.

[7] Md Yusof Z, Ibrahim Z, Ibrahim I, Mohd Azmi KZ, Ab. Aziz NA, Abd Aziz NH & Mohamad MS (2016), Distance Evaluated Simulated Kalman Filter for Combinatorial Optimization Problems, *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, No. 7, pp. 4904-4910.

[8] Muhammad B, Ibrahim Z, Mohd Azmi KZ, Abas KH, Ab. Aziz NA, Abd Aziz NH & Mohamad MS, Four Different Methods to Hybrid Simulated Kalman Filter (SKF) with Particle Swarm Optimization (PSO), *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 843-853.

[9] Muhammad B, Ibrahim Z, Ghazali KH, Mohd Azmi KZ, Ab. Aziz NA, Abd Aziz NH & Mohamad MS (2015), 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, pp. 17171-17176.

[10] Muhammad B, Ibrahim Z, Mohd Azmi KZ, Abas KH, Ab Aziz NA, Abd Aziz NH & Mohamad MS, Four Different Methods to Hybrid Simulated Kalman Filter (SKF) with Gravitational Search Algorithm (GSA), *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 854-864.

[11] Muhammad B, Ibrahim Z, Mat Jusof MF, Ab Aziz NA, Abd Aziz NH & Mokhtar N, A Hybrid Simulated Kalman Filter - Gravitational Search Algorithm (SKF-GSA), *International Conference on Artificial Life and Robotics*, (2017), pp. 707-710.

[12] Ab Rahman T, Ibrahim Z, Ab. Aziz NA, Zhao S & Abdul Aziz NH (2018), Single-Agent Finite Impulse Response Optimizer for Numerical Optimization Problems, *IEEE Access*, Vol. 6, pp. 9358-9374.

[13] Dogan B & Olmez T (2015), A New Metaheuristic for Numerical Function Optimization: Vortex Search Algorithm, *Inf. Sci. (Ny).*, Vol. 293, pp. 125-145.

[14] Abdul Aziz NH, Ibrahim Z, Ab. Aziz NA, Mohamad MS & Watada J (2018), Single-solution simulated Kalman filter algorithm for global optimisation problems, *Sadhana*, Vol. 123, No. 4, pp. 2333–2335.

[15] Mirjalili S, Mirjalili SM & Lewis A (2014), Grey Wolf Optimizer, *Adv. Eng. Softw.*, Vol. 69, pp. 46-61.

[16] Ab. Aziz NA, Mubin M, Ibrahim Z & Nawawi SW (2015), Statistical Analysis for Swarm Intelligence - Simplified, *Int. J. Futur. Comput. Commun.*, Vol. 4, No. 3, pp. 193-197, 2015.

[17] Liang JJ, Qu BY & Suganthan PN, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization (2013).

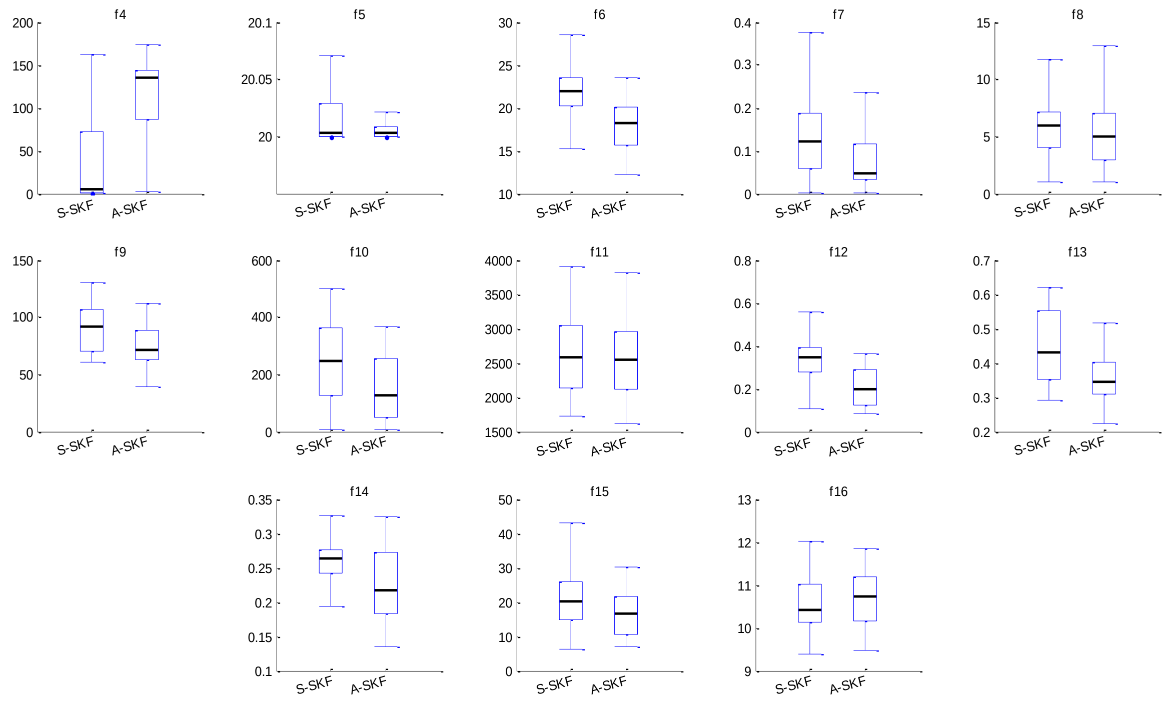

Fig. 4: Fitness error distribution of simple multimodal functions.

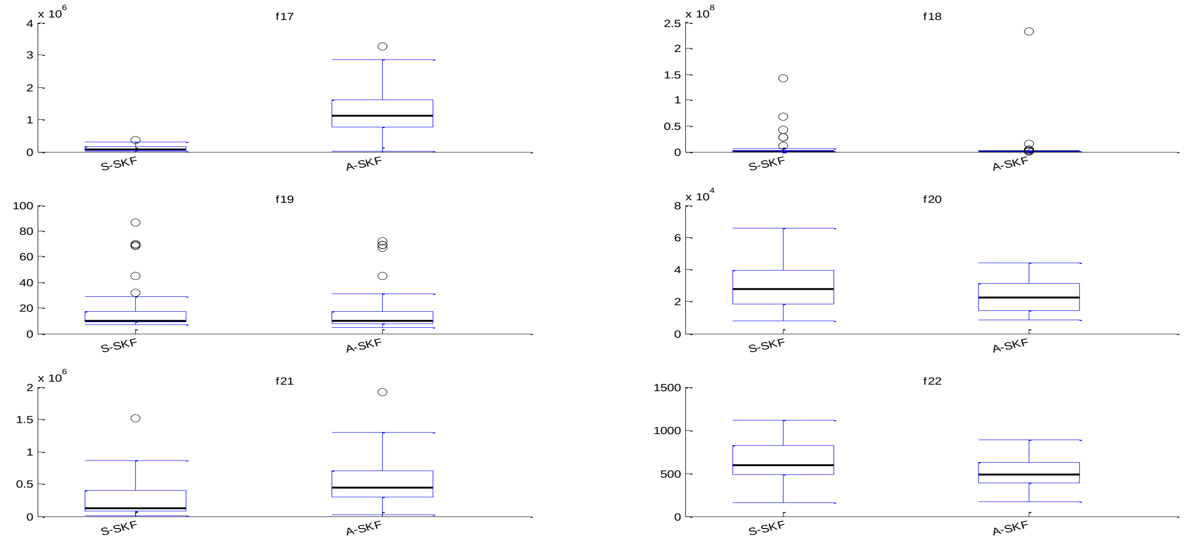

Fig. 5: Fitness error distribution of hybrid functions.

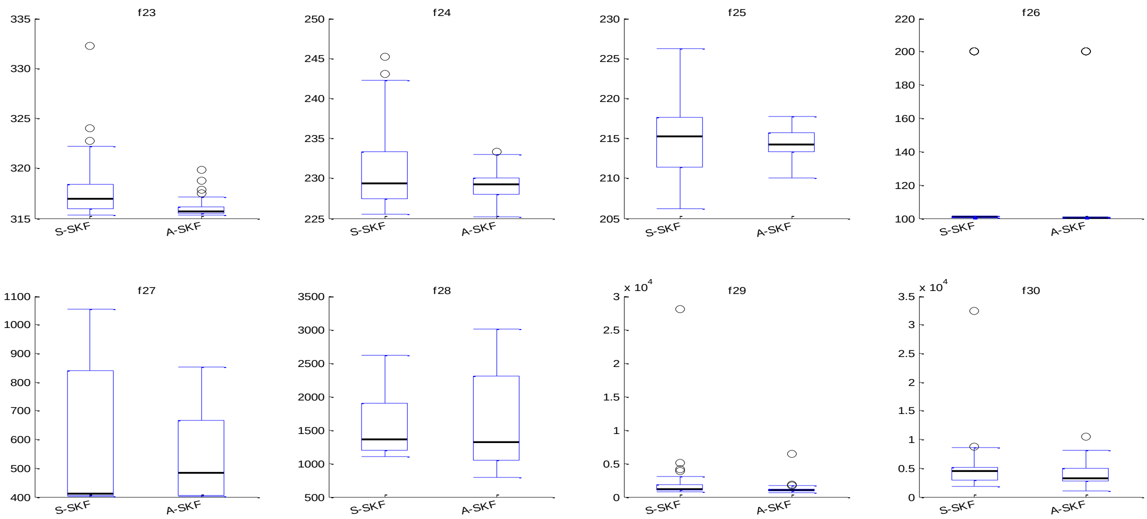

Fig. 6: Fitness error distribution of composite functions for S-SKF and A-SKF.

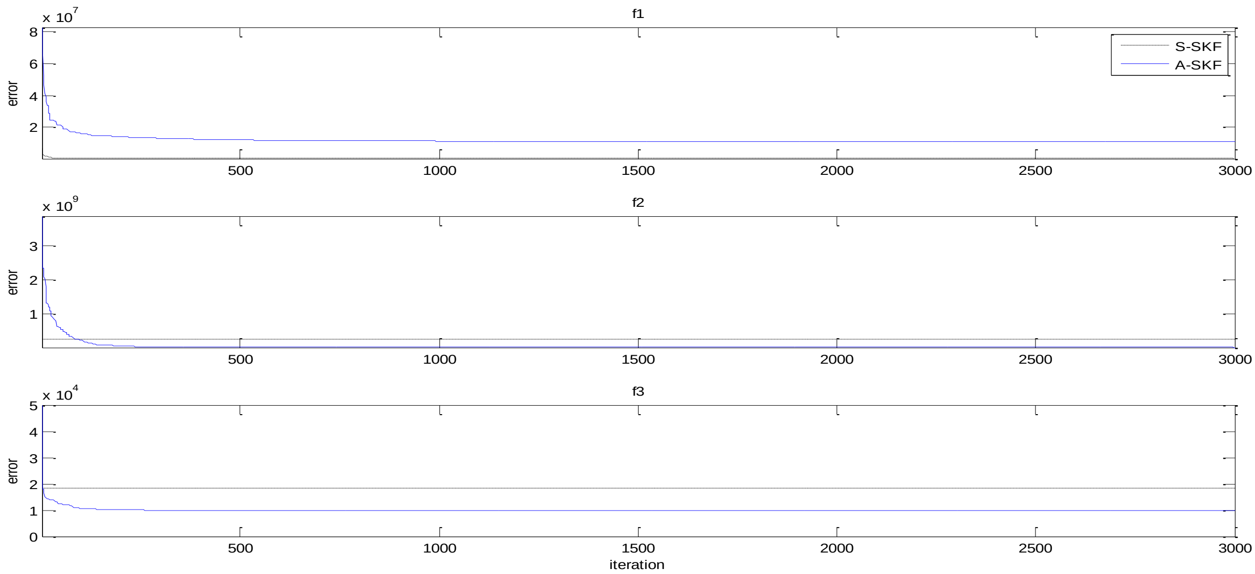

Fig. 7: Fitness error rate of unimodal functions.

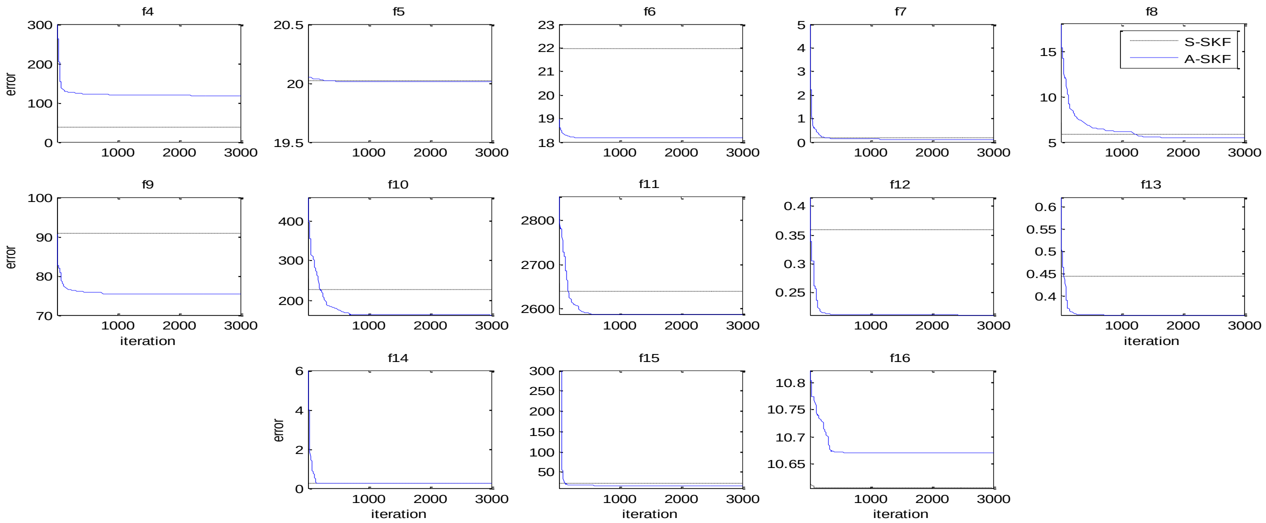

Fig. 8: Fitness error rate of simple multimodal functions.

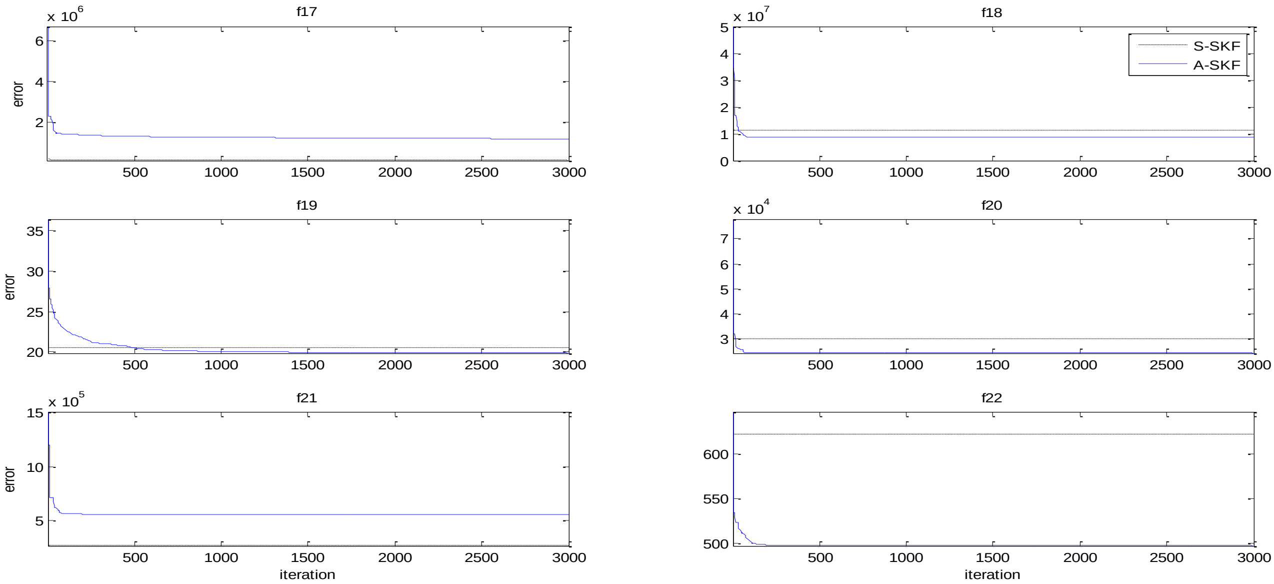

Fig. 9: Fitness error rate of hybrid functions.

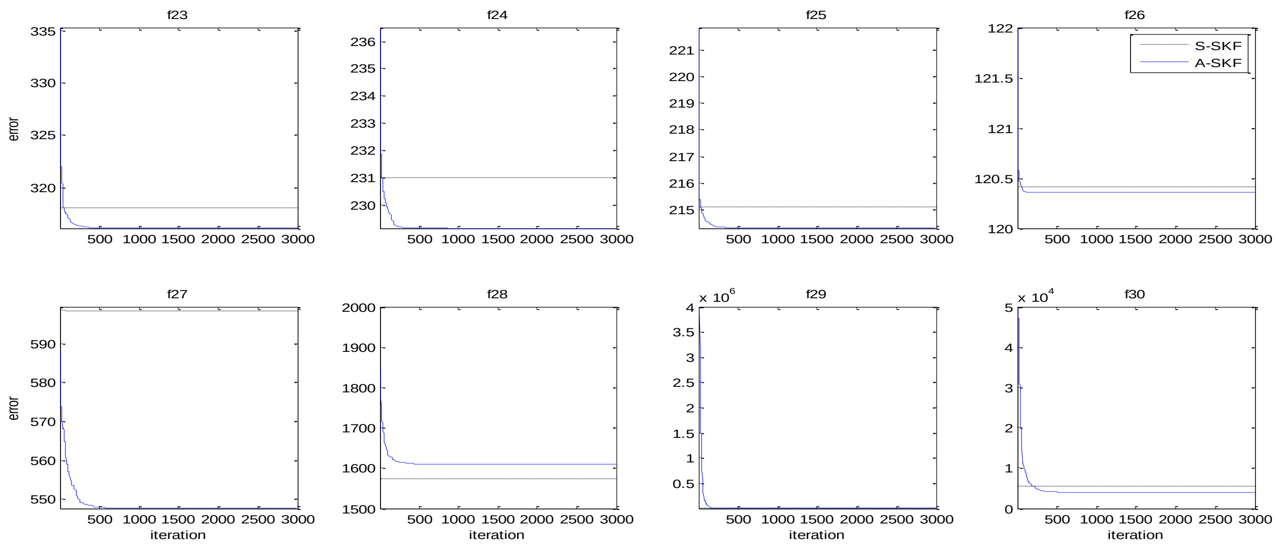

Fig. 10: Fitness error rate of composite functions.