A Hybrid Simulated Kalman Filter - Gravitational Search Algorithm (SKF-GSA)

Badaruddin Muhammad, Zuwairie Ibrahim, Mohd Falfazli Mat Jusof
Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia

Nor Azlina Ab Aziz, Nor Hidayati Abd Aziz
Faculty of Engineering and Technology, Multimedia University,
75450 Melaka, Malaysia

Norrima Mokhtar
Faculty of Engineering, University of Malaya,
50603 Kuala Lumpur, Malaysia

E-mail: badaruddinmuhammad@yahoo.com, zuwairie@ump.edu.my, mfalfazli@ump.edu.my, azlina.aziz@mmu.edu.my, hidayati.aziz@mmu.edu.my, norrimamokhtar@um.edu.my

Abstract

In this paper, simulated Kalman filter (SKF) and gravitational search algorithm (GSA) are hybridized in such a way that GSA is employed as prediction operator in SKF. The performance is compared using CEC2014 benchmark dataset. The proposed hybrid SKF-GSA shown to perform better than individual SKF and GSA algorithm.

Keywords: hybrid, simulated Kalman filter, gravitational search algorithm, CEC2014 benchmark problem.

1. Introduction

The simulated Kalman filter (SKF) and gravitational search algorithm (GSA) are examples of population-based optimization algorithms. GSA has been introduced in 2009 by Rashedi *et al*. [1]. On the other hand, as a new estimation-based metaheuristic [2], the SKF has been introduced by Ibrahim *et al.* [3] in 2015. Even though both algorithms are population-based, however, they are inspired differently. In particular, GSA is inspired by Newtonian law of gravity and law of motion while SKF is inspired by the estimation capability of Kalman filter. In this paper, hybridization between GSA and SKF is proposed. Specifically, GSA is employed during the prediction stage of SKF.

2. Simulated Kalman Filter Algorithm

The SKF algorithm is illustrated in Figure 1. Consider n number of agents, SKF algorithm begins with initialization of n agents, in which the states of each agent are given randomly. The maximum number of iterations, t_{max}, is defined. The initial value of error covariance estimate, $P(0)$, the process noise value, Q, and the measurement noise value, R, which are required in Kalman filtering, are also defined during initialization stage. Then, every agent is subjected to fitness evaluation to produce initial solutions $\{\boldsymbol{X}_1(0), \boldsymbol{X}_2(0), \boldsymbol{X}_3(0), \ldots, \boldsymbol{X}_{n-2}(0), \boldsymbol{X}_{n-1}(0), \boldsymbol{X}_n(0)\}$. The fitness values are compared and the agent having the best fitness value at every iteration, t, is registered as $\boldsymbol{X}_{\mathbf{best}}(t)$. For function minimization problem,

$$\boldsymbol{X_{best}}(t) = \min_{i \in 1,\ldots,n} fit_i(\boldsymbol{X}(t)) \quad (1)$$

The-best-so-far solution in SKF is named as $\boldsymbol{X}_{\mathbf{true}}$. The $\boldsymbol{X}_{\mathbf{true}}$ is updated only if the $\boldsymbol{X}_{\mathbf{best}}(t)$ is better (($\boldsymbol{X}_{\mathbf{best}}(t) < \boldsymbol{X}_{\mathbf{true}}$ for minimization problem, or $\boldsymbol{X}_{\mathbf{best}}(t) > \boldsymbol{X}_{\mathbf{true}}$ for maximization problem) than the $\boldsymbol{X}_{\mathbf{true}}$.

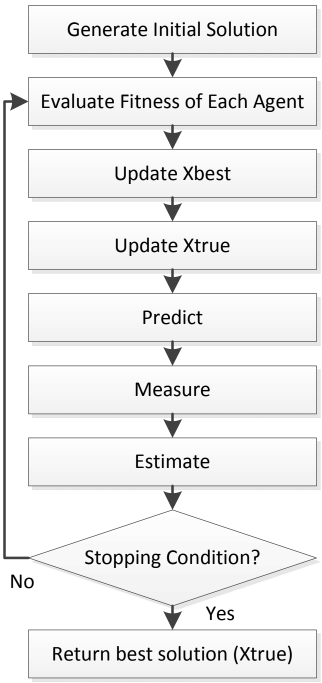

Fig. 1. The SKF algorithm.

The subsequent calculations are largely similar to the predict-measure-estimate steps in Kalman filter. In the prediction step, the following time-update equations are computed as follows:

$$\boldsymbol{X_i}(t|t) = \boldsymbol{X_i}(t) \quad (3)$$

$$P(t|t) = P(t) + Q \quad (4)$$

where $\boldsymbol{X}_i(t)$ and $\boldsymbol{X}_i(t|t)$ are the current state and transition/predicted state, respectively, and $P(t)$ and $P(t|t)$ are previous error covariant estimate and transition error covariant estimate, respectively. Note that the error covariant estimate is influenced by the process noise, Q.

The next step is measurement, which is modelled such that its output may take any value from the predicted state estimate, $\boldsymbol{X}_i(t|t)$, to the true value, $\boldsymbol{X}_{true}$. Measurement, $\boldsymbol{Z}_i(t)$, of each individual agent is simulated based on the following equation:

$$\boldsymbol{Z_i}(t) = \boldsymbol{X_i}(t|t) + \sin(rand \times 2\pi) \times |\boldsymbol{X_i}(t|t) - \boldsymbol{X_{true}}| \quad (5)$$

The $\sin(rand \times 2\pi)$ term provides the stochastic aspect of SKF algorithm and $rand$ is a uniformly distributed random number in the range of $[0,1]$.

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

$$K(t) = \frac{P(t|t)}{P(t|t) + R} \quad (6)$$

Then, the estimation of next state, $\boldsymbol{X}_i(t)$, is computed based on Eqn. (7).

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t|t) + K(t) \times (\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t)) \quad (7)$$

and the error covariant is updated based on Eqn. (8).

$$P(t+1) = \big(1 - K(t)\big) \times P(t|t) \quad (8)$$

Finally, the next iteration is executed until the maximum number of iterations, $t_{\max}$, is reached.

3. Gravitational Search Algorithm

In GSA, agents are considered as an object and their performance are expressed by their masses. The position of particle is corresponding to the solution of the problem. $\text{best}(t)$ and $\text{worst}(t)$ denote the best and the worst fitness value of the population t. The *best* and the *worst* for the case of function minimization problem are defined as follows:

$$\begin{aligned} best(t) &= \min_{j \in \{1,\ldots,N\}} fit_j(t) \\ worst(t) &= \max_{j \in \{1,\ldots,N\}} fit_j(t) \end{aligned} \quad (12)$$

while gravitational constant is defined as a decreasing function of time, which is set to G_0 at the beginning and decreases exponentially towards zero with lapse of time.

To give a stochastic characteristic to GSA, the total force acted on agent i in dth dimension is a randomly weighted sum of dth components of the forces exerted from other agents.

According to law of motion, the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Acceleration of any mass is equal to the force acted on the system divided by mass of inertia. Finally, the next iteration is executed until the maximum number of iterations, $t_{\max}$, is reached. In summary, the algorithm of standard GSA is shown in Figure 2.

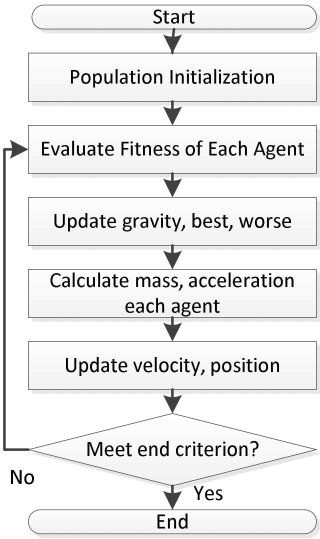

Fig. 2. The GSA algorithm.

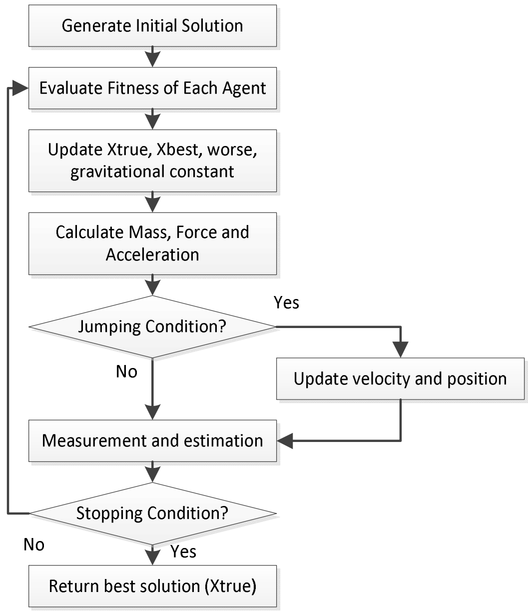

Fig. 3. Hybrid SKF-GSA algorithm.

4. Hybrid SKF-GSA Algorithm

Note that even though the SKF follows predict-measure-estimate steps as in Kalman filter, the states are not updated during the predict step. Hence, in the proposed hybrid SKF-GSA algorithm, GSA is employed as the prediction operator in SKF. An additional variable is introduced in hybrid SKF-GSA, which is the jumping rate, J_r, that is a predefined constant in the range of [0,1]. Prediction based on GSA is performed if jumping rate condition is satisfied. The hybrid SKF-GSA algorithm is shown in Figure 3.

In detail, the hybrid SKF-GSA algorithm begins with initialization of n agents, in which the states of each agent are given randomly. The maximum number of iterations, t_{max}, the initial value of error covariance estimate, $P(0)$, the process noise value, Q, the measurement noise value, R, and jumping rate value, J_r, are also defined during initialization stage. Then, every agent is subjected to fitness evaluation to produce initial solutions. After that, $\boldsymbol{X}_{\mathbf{best}}(t)$ and $\boldsymbol{X}_{\mathbf{true}}$ are updated according to SKF algorithm and *pbest* is updated according to GSA algorithm.

In hybrid SKF-GSA, the purpose of jumping rate, J_r, is to control the occurrence of the prediction. Based on our observation, the performance of SKF cannot be enhanced when GSA is executed at every iteration as the prediction operator of SKF. The following jumping condition is considered:

```
if   rand < Jr
     apply GSA in prediction
else
     proceed to measurement and estimation
end
```

where *rand* is a random number in the range of [0,1]. If $rand < J_r$, agents' velocity is updated according to GSA. For the position update, $\boldsymbol{X}_{\mathbf{predict}}$, is required and it is calculated as follows:

$$\boldsymbol{x}_{\boldsymbol{predict}}(t) = x_i(t) + v_i(t+1) \tag{12}$$

The algorithm continues with measurement and estimation similar to SKF. The next iteration is executed until the maximum number of iterations, t_{max}, is reached.

5. Experiment, Result, and Discussion

The CEC2014 benchmark functions (http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm) have been employed for performance evaluation. Table 1 shows the setting parameters used in experiments.

The experimental result for CEC2014 benchmark functions are tabulated in Table 2. Result in bold represents the best performance.

Table 1. Setting Parameters

Experimental Parameters	
Number of agent	100
Number of dimension	50
Number of run	50
Number of iteration	10,000
Search space	[-100.100]
SKF Parameters	
Error covariance estimate, P	1000
Process noise value, Q	0.5
Measurement noise value, R	0.5
GSA Parameters	
α	20
Initial gravitational constant, G_o	100
SKF-GSA Parameters	
Jumping rate, J_r	0.1

Table 2. The Average Fitness Values Obtained by SKF, GSA, and SKF-GSA

Function	SKF	SKF-GSA	GSA
F1	4702013.17	4295218.33	**1400195.11**
F2	24498691.7	8265.524	**7166.7916**
F3	18147.7005	**16725.114**	64249.0520
F4	**532.77148**	538.979	653.4553
F5	520.010016	519.9999	**519.9997**
F6	633.441686	**627.7067**	636.3361
F7	700.246225	700.0133	**700.00014**
F8	**807.981323**	816.2199	1076.43847
F9	1059.13877	**1056.2279**	1217.9866
F10	**1335.18324**	1543.5057	7456.3162
F11	6249.36725	**6178.1897**	8637.6403
F12	1200.23641	1200.0562	**1200.0001**
F13	1300.55973	1300.5183	**1300.3749**
F14	1400.30009	**1400.2931**	1400.3012
F15	1551.6584	1549.9115	**1504.4108**
F16	**1619.12553**	1619.2702	1622.5712
F17	908272.092	880075.612	**161088.839**
F18	6941389.77	**3285.3104**	3731.2078
F19	1950.223	1950.84943	**1923.7518**
F20	34799.058	**25328.3173**	26574.2424
F21	1186640.91	1092401.31	**187636.63**
F22	3429.1058	**3339.9376**	3857.9572
F23	2645.6890	2644.6327	**2500.0000**
F24	2667.2498	2660.3817	**2600.0283**
F25	2730.4018	2731.4575	**2700.0000**
F26	**2766.3853**	2786.2986	2800.0315
F27	3883.3415	**3763.4039**	4577.5301
F28	7223.3697	7757.2433	**6261.3240**
F29	5997.8302	4109.6202	**3100.1482**
F30	19753.2888	18853.2102	**8695.4410**

Based on the averaged performances, Wilcoxon signed rank test is performed and the result is tabulated in Table 3. Based on the level of significant, $\sigma = 0.05$, it is found that statistically, the proposed SKF-GSA is significantly superior to SKF and GSA in solving continuous numerical optimization problems.

Table 3. Wilcoxon Signed-Rank Test Result

Comparison	R^-	R^+
Hybrid SKF-GSA vs GSA	219	246
Hybrid SKF-GSA vs SKF	361	104

6. Conclusion

This paper report an attempt to hybrid SKF algorithm with a well-established GSA algorithm. In this study, GSA is chosen as the prediction mechanism in SKF algorithm. In addition, jumping rate is also incorporated in the proposed SKF-GSA algorithm. During the prediction, GSA is executed not only when the jumping rate condition is satisfied but also if the predicted solution is better. The findings proved that the proposed hybrid SKF-GSA is superior to individual SKF and GSA algorithms

Acknowledgements

This research is supported by a Fundamental Research Grant Scheme (FRGS) awarded to Universiti Malaysia Pahang (RDU160105).

References

1. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, *Information Sciences.* **179** (13) (2009) 2232-2248.
2. N. H. Abdul Aziz, Z. Ibrahim, S. Razali, N. A. Ab. Aziz, Estimation-based metaheuristics: a new branch of computational intelligence, in *Proc. 3rd National Conf. Postgraduate Research* (Pekan, Malaysia, 2016), pp. 469-476.
3. Z. Ibrahim, N. H. Abdul Aziz, N. A. Ab. Aziz, S. Razali, M. I. Shapiai, S. W. Nawawi, M. S. Mohamad, A Kalman filter approach for solving unimodal optimization problems, *ICIC Express Letters.* **9**(12) (2015) 3415-3422.