A NEW HYBRID SIMULATED KALMAN FILTER AND PARTICLE SWARM OPTIMIZATION FOR CONTINUOUS NUMERICAL OPTIMIZATION PROBLEMS

Badaruddin Muhammad[1], Zuwairie Ibrahim[1], Kamarul Hawari Ghazali[1], Kamil Zakwan Mohd Azmi[1], Nor Azlina Ab Aziz[2], Nor Hidayati Abd Aziz[1,2], and Mohd Saberi Mohamad[3]
[1]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, Malaysia
[2]Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia
[3]Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia
E-Mail: badaruddinmuhammad@yahoo.com

ABSTRACT
Inspired by the estimation capability of Kalman filter, we have recently introduced a novel population-based optimization algorithm called simulated Kalman filter (SKF). Every agent in SKF is regarded as a Kalman filter. Based on the mechanism of Kalman filtering, which includes prediction, measurement, and estimation, the global minimum/maximum can be estimated. Measurement process, which is required in Kalman filtering, is mathematically modelled and simulated. Agents communicate among them to update and improve the solution during the search process. Inspired by the bird flocking, particle swarm optimization (PSO) has been introduced in 1994. In PSO, a swarm of agent search the global minimum/maximum by velocity and position updates, which are influenced by current position of agent, current position of agent, personal best, and global best of the swarm. In this research, SKF and PSO are hybridized in such a way that PSO is employed as prediction operator in SKF. The performance of the proposed hybrid SKF-PSO algorithm (SKF-PSO) is compared against SKF and PSO using CEC2014 benchmark dataset for continuous numerical optimization problems. Based on the analysis of experimental results, we found that the proposed hybrid SKF-PSO is superior to both SKF and PSO algorithm.

Keywords: simulated kalman filter, particle swarm, optimization, and cec2014 benchmark problem.

INTRODUCTION

The main objective of an optimization problem is to find the best combination of real-valued variables of a fitness function such that the value of the fitness is maximum or minimum. This can be achieved efficiently by employing a population-based optimization algorithm.

The simulated Kalman filter (SKF) and particle swarm optimization (PSO) are examples of population-based optimization algorithms. PSO has been introduced in 1994 by Eberhart and Kennedy [1] while SKF has been recently introduced by Ibrahim *et al.* [2] in 2015. Even though both algorithms are population-based, however, they are inspired differently. In particular, PSO is inspired by bird flocking behaviour while SKF is inspired by the estimation capability of Kalman filter.

In literature, PSO has been subjected to various improvements including hybridization with other optimization algorithms. For example, PSO can be hybridized with gravitational search algorithm [3-4], chemical reaction optimization [5], differential evolution [6], and Extremal optimization [7].

In this research, hybridization between PSO and a recently introduced SKF is proposed. From the SKF point of view, this is among the first attempt to improve its performance by hybridization with other algorithm such as PSO.

This paper is organized as follows. At first, SKF and PSO algorithms will be reviewed. After that, the new hybrid SKF-PSO will be explained in detail. Experimental procedure will be presented and the superiority of the new hybrid SKF-PSO will be shown and discussed. Lastly, conclusion will be given at the end of this paper.

SIMULATED KALMAN FILTER ALGORITHM

The simulated Kalman filter (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 $\boldsymbol{X}_{best}(t)$. For function minimization problem,

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

whereas, for function maximization problem,

$$\boldsymbol{X}_{best}(t) = \max_{i\in 1,\ldots,n} fit_i(\boldsymbol{X}(t)) \quad (2)$$

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

VOL. 10, NO. 22, DECEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

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.

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

$$\boldsymbol{Z}_i(t) = \boldsymbol{X}_i(t|t) + \sin(rand \times 2\pi) \times |\boldsymbol{X}_i(t|t) - \boldsymbol{X}_{true}| \quad (4)$$

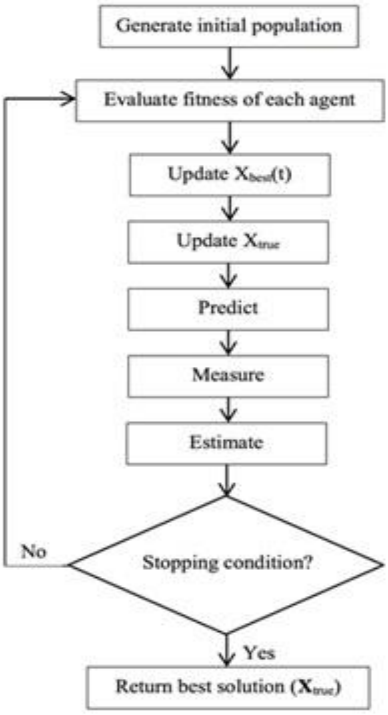

Figure-1. The simulated Kalman filter (SKF) algorithm.

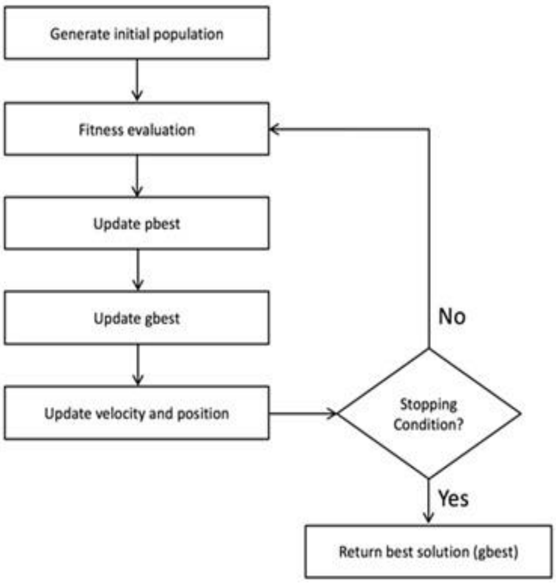

Figure-2. The particle swarm optimization (PSO) algorithm.

where $\boldsymbol{X}_i(t)$ and $\boldsymbol{X}_i(t|t)$ are the previous 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 a feedback to estimation process. Measurement 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, $\boldsymbol{X}_i(t+1)$, is computed based on Equation. (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 Equation. (8).

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

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

PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) is illustrated in Figure-2. Consider n number of particle, PSO begins with initialization of n particles, in which the coordinates of ith particle, $x_i(0)$, are given randomly. The maximum number of iterations, $t_{\max}$, and initial velocity of ith particle, $\boldsymbol{v}_i(0)$, are also defined during the initialization.

Then, every particle is subjected to fitness evaluation to produce initial solutions. Personal best, *pbest*, and global best, *gbest*, are updated. After that, the velocity and position are updated as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[pbest - x_i(t)] + c_2 r_2[gbest - x_i(t)] \quad (9)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (10)$$

where $v_i(t)$ and $v_i(t+1)$ are current and updated velocity, respectively, $x_i(t)$ and $x_i(t+1)$ are current and updated position, respectively, ω is called inertia weight, and r_1 and r_2 are called cognitive coefficient and social coefficient, respectively, which are uniformly distributed random numbers in the range of [0,1]. Lastly, the next iteration is executed until the maximum number of iterations, $t_{\max}$, is reached.

HYBRID SKF-PSO 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, as shown in Equation. (3). Hence, in the proposed hybrid SKF-PSO algorithm, PSO is employed as the prediction operator in SKF. An additional variable is introduced in hybrid SKF-PSO, which is the jumping rate, J_r, that is a predefined constant in the range of [0,1]. Prediction based on PSO is performed if jumping rate condition is satisfied. Once jumping rate condition is satisfied, fitness evaluation is performed again after the velocity is updated and next position is predicted. Then, agents move to the predicted position if better solution is found at the predicted position. The hybrid SKF-PSO algorithm is shown in Figure-3.

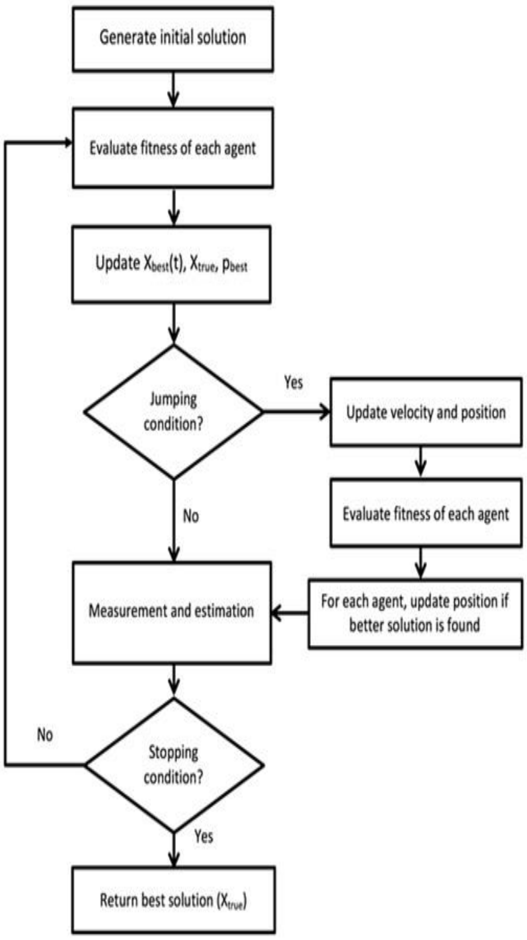

Figure-3. The new hybrid SKF-PSO algorithm.

In detail, the hybrid SKF-PSO 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 PSO algorithm.

In hybrid SKF-PSO, 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 PSO is executed at every iteration as the prediction operator of SKF. The following jumping condition is considered:

```
if  rand < Jr
    then
        apply PSO in prediction
    else
        proceed to measurement and estimation
else
```

where *rand* is a random number in the range of [0,1]. If *rand* < J_r, agents' velocity is updated as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[pbest - x_i(t)] + c_2 r_2[X_{true} - x_i(t)] \quad (11)$$

Note that the velocity update is almost similar to PSO (Equation. (9)). The only different is that the *gbest* is replaced with $\boldsymbol{X}_{\mathbf{true}}$.

For the position update, agent moves to a new position only if the fitness of the new position is better than the fitness of the current position. Thus, pre-calculation of next position, which is called $\boldsymbol{X}_{\mathbf{predict}}$, is required and it is calculated as follows:

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

Then, the position is updated as follows:

$$\boldsymbol{x}_i(t+1) = \begin{cases} \boldsymbol{x}_{\mathbf{predict}}(t) & \text{if fit}\,(\boldsymbol{x}_{\mathbf{predict}}(t)) < fit\,(\boldsymbol{x}_i(t)) \\ \boldsymbol{x}_i(t) & \text{if fit}\left(\boldsymbol{x}_{\mathbf{predict}}(t)\right) > fit\,(\boldsymbol{x}_i(t)) \end{cases} \quad (13)$$

The algorithm continues with measurement and estimation similar to SKF using Equation. (5) to Equation. (8). The next iteration is executed until the maximum number of iterations, $t_{\max}$, is reached.

EXPERIMENTS

In this study, CEC2014 benchmark functions [8] have been employed for performance evaluation of the proposed hybrid SKF-PSO algorithm. Thirty functions are available, which consist of three unimodal functions, 13

VOL. 10, NO. 22, DECEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

multimodal functions, six hybrid functions, and eight composition functions, as shown in Table-1.

Table-1. The CEC2014 benchmark problems.

Function ID	Type	Ideal Fitness
F1	Unimodal	100
F2		200
F3		300
F4	Multimodal	400
F5		500
F6		600
F7		700
F8		800
F9		900
F10		1000
F11		1100
F12		1200
F13		1300
F14		1400
F15		1500
F16		1600
F17	Hybrid	1700
F18		1800
F19		1900
F20		2000
F21		2100
F22		2200
F23	Composition	2300
F24		2400
F25		2500
F26		2600
F27		2700
F28		2800
F29		2900
F30		3000

Table-2 shows the setting parameters used in experiment. The search space for all the test functions is [-100,100] for all dimensions. For PSO, linearly decreasing inertia weight is used which can be calculated as:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t \tag{14}$$

RESULT AND DISCUSSION

The experimental result for CEC2014 benchmark functions are tabulated in Table-3. Result in bold represent the best performance. It is found that the proposed hybrid SKF-PSO performed better than SKF and PSO in most problems, particularly, 17 functions out of 30. Examples of convergence curves are shown in Figure-4 to Figure-6.

Table-2. Setting parameters.

Experimental Parameters	
Number of agent	100
Number of dimension	50
Number of run	50
Number of iteration	2000
Search space	[-100.100]
rand	[-1,1]
SKF Parameters	
Error covariance estimate, P	1000
Process noise, Q	0.5
Measurement noise, R	0.5
PSO Parameters	
$\omega_{\max}$	0.9
$\omega_{\min}$	0.1
Cognitive coefficient, c_1	2
Social coefficient, c_2	2
SKF-PSO Parameters	
Jumping rate, J_r	0.2

Figure-4. Convergence curve for function F2.

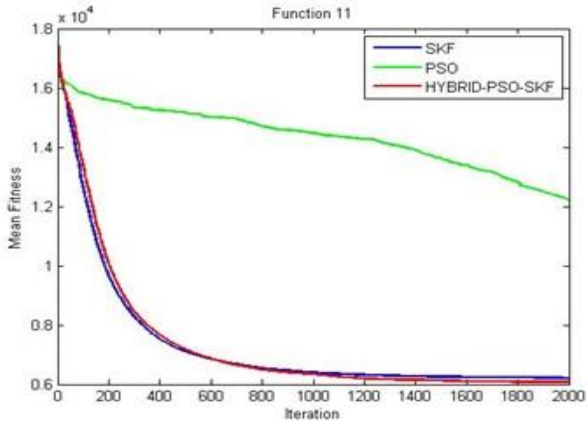

Figure-4. Convergence curve for function F11.

Table-3. The average fitness value obtained by SKF, hybrid SKF-PSO, and PSO algorithms. Numbers in bold indicate the best fitness.

Function	SKF	SKF-PSO	PSO
F1	17369876.77	**14888456.82**	44394120.17
F2	18365308.17	**9308.25**	25530051.35
F3	16118.04	**15192.94**	28696.06
F4	626.23	**620.70**	836.66
F5	**520.01**	520.06	521.10
F6	631.96	**631.19**	633.50
F7	701.26	700.09	**700.01**
F8	822.54	**819.01**	894.84
F9	**1059.57**	1060.54	1076.70
F10	1426.18	**1386.46**	2372.90
F11	6203.75	**6075.93**	12218.77
F12	**1200.25**	1200.54	1202.94
F13	1300.56	**1300.53**	1300.57
F14	**1400.3**	1400.30	1400.38
F15	1556.67	1534.90	**1531.00**
F16	**1619.43**	1619.51	1622.04
F17	2816604.65	**1833022.97**	3477529.11
F18	8997221.49	**14744.00**	277318.63
F19	1958.34	**1946.80**	1960.50
F20	35668.02	**13214.16**	18579.12
F21	3111583.02	**516879.54**	859993.75
F22	3473.36	3352.37	**3264.90**
F23	2649.31	**2647.11**	2654.93
F24	2666.49	**2664.14**	2676.79
F25	2731.71	**2720.80**	2729.80
F26	2792.91	2735.52	**2700.54**
F27	**3905.2**	4352.18	3940.92
F28	**6934.64**	8847.23	6938.08
F29	19573.46	379376.23	**4760.56**
F30	**25820.54**	233143.69	116075.51

Table-4. Wilcoxon test result.

Comparison	R^-	R^+
Hybrid PSO-SKF vs PSO	335	130
Hybrid PSO-SKF vs SKF	355	110

Based on the averaged performances, Wilcoxon signed rank test is performed. The result of the test is tabulated in Table 4. The level of significant chosen here is $\sigma = 0.05$. It is found that statistically, the proposed SKF-PSO algorithm not only performed better than SKF and PSO algorithm in most problems, SKF-PSO also significantly superior than SKF and PSO in solving continuous numerical optimization problems.

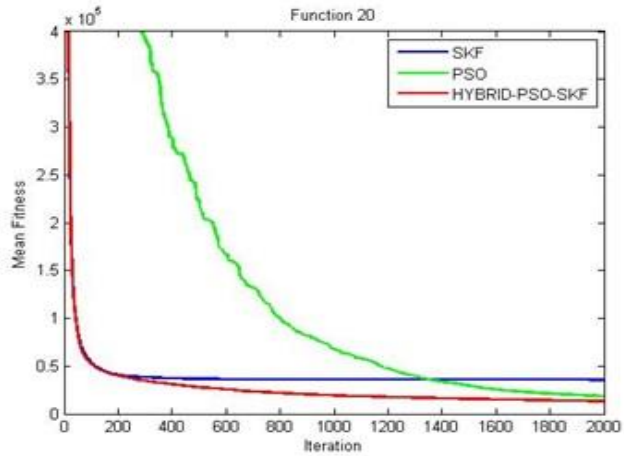

Figure-5. Convergence curve for function F20.

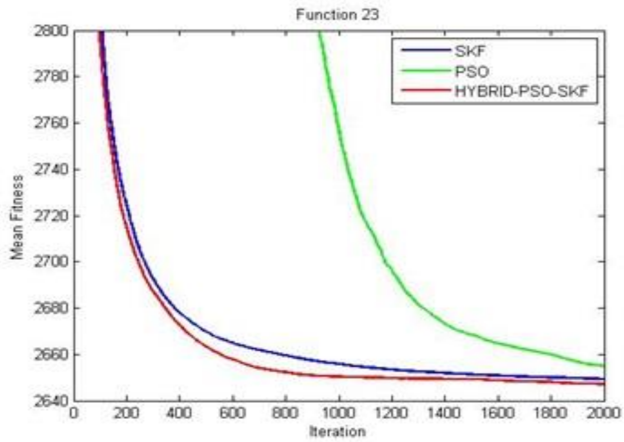

Figure-6. Convergence curve for function F23.

CONCLUSION

This paper report the first attempt to hybrid a recently introduced SKF algorithm with a well-established in PSO algorithm. In this study, PSO is chosen as the prediction mechanism in SKF algorithm. In addition, jumping rate is also incorporated in the proposed SKF-PSO algorithm. During the prediction, PSO 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-PSO is superior to SKF and PSO algorithms. Currently, more experiments are being done. Another well-established swarm intelligence algorithm is considered to be prediction operator in SKF.

ACKNOWLEDGEMENTS

This work is financially supported by the UMP Post Graduate Research Scheme (GRS1503120) and by the Fundamental Research Grant Scheme (FRGS/1/2015/TK04/MMU/03/2) awarded by the Ministry of Higher Education (MOHE) to Multimedia University (MMU).

REFERENCES

[1] Kennedy J. and Eberhart R. C. 1995. Particle swarm optimization. IEEE International Conference on Neural Networks, Vol. IV, pp. 1942-1948.

[2] Ibrahim Z., Abdul Aziz N. H., Ab Aziz N. A., Razali S., Shapiai M. I., Nawawi S. W., and Mohamad M. S. A Kalman filter approach for solving unimodal optimization problems. 2015. ICIC Express Letters (accepted).

[3] Azali S. and Sheikhan M. 2015. Intelligent control of photovoltaic system using BPSO-GSA-optimized neural network and fuzzy-based PID for maximum power point tracking. Applied Intelligence. July 2015.

[4] Purcaru C., Precup R-E, Iercan D., Fedorovici L.-O., and David R.-C. 2013. Hybrid PSO-GSA robot path planning algorithm in static environments with danger zones. 17th International Conference on System Theory, Control and Computing, pp. 434-439.

[5] Li Z., Nguyen T. T., Chen, S. M., and Truong T. K. 2015. A hybrid algorithm based on particle swarm and chemical reaction optimization for multi-object problems. Applied Soft Computing. Vol. 35, pp. 525-540.

[6] Khamsawang S., Wannakarn P., and Jiriwibhakorn S., 2010. Hybrid PSO-DE for solving the economic dispatch problem with generator constraints. The 2nd International Conference on Computer and Automation Engineering, Vol. 5, pp. 135-139.

[7] Zhenyi C., Gaofeng W., and Chen D. 2012. An adaptive hybrid combination of PSO and Extremal Optimization. 2012. International Conference on Information Science and Technology, pp. 712-715. http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014.