International Journal of Engineering & Technology, 7 (4.27) (2018) 22-29

International Journal of Engineering & Technology

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

Research paper

Distance Evaluated Simulated Kalman Filter with State Encoding for Combinatorial Optimization Problems

Zulkifli Md Yusof[1]*, Zuwairie Ibrahim[2], Asrul Adam[3], Kamil Zakwan Mohd Azmi[4], Tasiransurini Ab Rahman[5], Badaruddin Muhammad[6], Nor Azlina Ab Aziz[7], Nor Hidayati Abd Aziz[8], Norrima Mokhtar[9], Mohd Ibrahim Shapiai[10], Mohd Saberi Muhammad[11]

[1]Faculty of Manufacturing, Universiti Malaysia Pahang, Malaysia.
[2]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Malaysia.
[3]Faculty of Engineering and Technology, Multimedia University, Malaysia
[4]Faculty of Engineering, University Malaya, Malaysia
[5]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Malaysia.
[6]Institute for Artificial Intelligence and Big Data, Faculty of Bioengineering and Technology, Universiti Malaysia Kelantan, Malaysia.
**Corresponding author E-mail: zuwairie@ump.edu.my*

Abstract

Simulated Kalman Filter (SKF) is a population-based optimization algorithm which exploits the estimation capability of Kalman filter to search for a solution in a continuous search space. The SKF algorithm only capable to solve numerical optimization problems which involve continuous search space. Some problems, such as routing and scheduling, involve binary or discrete search space. At present, there are three modifications to the original SKF algorithm in solving combinatorial optimization problems. Those modified algorithms are binary SKF (BSKF), angle modulated SKF (AMSKF), and distance evaluated SKF (DESKF). These three combinatorial SKF algorithms use binary encoding to represent the solution to a combinatorial optimization problem. This paper introduces the latest version of distance evaluated SKF which uses state encoding, instead of binary encoding, to represent the solution to a combinatorial problem. The algorithm proposed in this paper is called state-encoded distance evaluated SKF (SEDESKF) algorithm. Since the original SKF algorithm tends to converge prematurely, the distance is handled differently in this study. To control and exploration and exploitation of the SEDESKF algorithm, the distance is normalized. The performance of the SEDESKF algorithm is compared against the existing combinatorial SKF algorithm based on a set of Traveling Salesman Problem (TSP).

Keywords: *combinatorial optimization; distance evaluated; simulated Kalman filter; state encoding; travelling salesman problem*

1. Introduction

Most optimization problems in their original form can be categorized as numerical optimization or combinatorial optimization problems. The numerical optimization problems involve finding the best possible numerical values to a set of parameters. Examples of numerical optimization problems are Proportional-Integral-Derivative (PID) tuning in control applications [1], training of a neural network [2], adaptive beamforming [3], and engineering design problems [4]. On the other hand, combinatorial optimization or discrete problems concern with the best combination of a set of variables such as in traveling salesman problems [5], assembly sequence planning [6], scheduling for airport [7] and classroom [8], routing in integrated circuits [9], and feature selection [10].

In solving discrete and combinatorial optimization problems, metaheuristic algorithms such genetic algorithm (GA) [11] and ant colony optimization (ACO) [12] have been developed to operate in binary search space. However, not all optimization algorithms are originally developed to operate in binary search space. An example of this algorithm is simulated Kalman filter (SKF) [13-14].

The SKF algorithm is inspired by the estimation capability of Kalman filtering [15]. In SKF algorithm, every agent is regarded as a Kalman filter. Based on the mechanism of Kalman filtering and measurement process, every agent estimates the global minimum/maximum in a search space. Measurement, 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.

To date, the SKF algorithm has been applied to solve several engineering problems. In signal processing, Adam *et al*. has employed angle-modulated SKF as feature selection in peak classification of EEG signal [16-17]. In telecommunication engineering, the SKF algorithm has been used as adaptive beamforming algorithm [18-21]. In industrial engineering, the SKF has been used to solve printed circuit board drill path optimization problem [22-23] and assembly sequence planning problem [24]. In scheduling application, the SKF has been employed in solving airport gate allocation problem [25-26]. In image processing, the SKF has been used as a template matching algorithm in distance measurement [27-28]. In system identification, the SKF algorithm has been used to estimate the model order and parameter value of an ARX model [29-30]. The SKF algorithm also has been introduced as a tuning method for proportional-integral-derivative (PID) controller [31].

Fundamentally, studies of the SKF algorithm have been reported [32-33]. Furthermore, modifications of the SKF [34-35] and hybridization with other algorithms [36-40] have been done to further improve the performance of the SKF algorithm.

The original SKF algorithm only capable to solve numerical optimization problems which involves continuous search space. Combinatorial optimization problem, on the other hand, is also an important study in optimization. Normally, the combinatorial optimization problem requires the search space to be in discrete form, which is either 0 or 1. To enable the SKF to operate in discrete search space, binary SKF (BSKF) [41], distance evaluated SKF (DESKF) [42-43], and angle modulated SKF (AMSKF) [44] have been proposed, previously.

In this paper, state-encoded distance evaluated SKF (SEDESKF) algorithm is proposed which search the solutions of a combinatorial optimization problem in state, instead of in binary number. The concept of state encoding in solving combinatorial optimizations is not totally new. This concept has been used in particle swarm optimization (PSO) [45-47] and gravitational search algorithm (GSA) [48-53], which used the velocity variable in PSO and GSA in updating the states. In this study, the distance variable is derived from the SKF algorithm in updating the states since the SKF algorithm operates without a velocity variable. Similar to the previous combinatorial algorithms using state encoding, the proposed SEDESKF requires less number of dimensions and consequently, less computations compared to the combinatorial optimization algorithms using binary encoding.

2. The Simulated Kalman Filter

Every agent in SKF is regarded as a Kalman filter. Based on the mechanism of Kalman filtering and measurement process, every agent estimates the global minimum/maximum. Measurement, which is required in Kalman filtering, is mathematically modelled and such simulated. Agents communicate among them to update and improve the solution during the search process. 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, $P(0) \in [0,1]$, the process noise covariance value, $Q \in [0,1]$, and the measurement noise covariance value, $R \in [0,1]$, 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 $\boldsymbol{X}_{best}(t)$.

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 than the $\boldsymbol{X}_{true}$ ($\boldsymbol{X}_{best}(t) < \boldsymbol{X}_{true}$ for function minimization problem, or $\boldsymbol{X}_{best}(t) > \boldsymbol{X}_{true}$ for maximization problem). The subsequent calculations are similar to the predict-measure-estimate steps in Kalman filtering. In the prediction step, the following time-update equations are computed.

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

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

where $\boldsymbol{X}_i(t)$ and $\boldsymbol{X}_i(t|t+1)$ denote the current state and current transition/predicted state, respectively, while the $\boldsymbol{P}_i(t)$ 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.

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, $\boldsymbol{X}_i(t|t+1)$, to the true value, $\boldsymbol{X}_{true}$. Measurement, $\boldsymbol{Z}_i(t)$, of each individual agent is simulated based on the following equation:

Table 1: Dimension number and dimension value for an instance of $\Sigma_8 = \{0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\}$

Dimension number	Dimension value
1	0
2	0
3	1
4	1
5	1
6	0
7	0
8	1
9	1

$$\boldsymbol{Z}_i(t) = \boldsymbol{X}_i(t|t+1) + \sin(2\pi r_i(t)) \times |\boldsymbol{X}_i(t|t+1) - \boldsymbol{X}_{true}| \tag{3}$$

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

The final step is estimate. During this step, Kalman gain, $K(t) \in [0,1]$ is computed as follows:

$$K(t) = P(t|t+1)/(P(t|t+1) + R) \tag{4}$$

Then, the estimation of next state, $\boldsymbol{X}_i(t+1)$, and the updated error covariant, $P_i(t+1)$, are computed based on (5) and (6), respectively.

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t|t+1) + K(t) \times |\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t+1)| \tag{5}$$

$$P(t+1) = (1 - K(t)) \times P(t|t+1) \tag{6}$$

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

3. The Existing Combinatorial Simulated Kalman Filter Algorithms

Previously, three extensions of SKF algorithm for combinatorial optimization problems have been introduced in literature. These extensions enable agent i in SKF to update a bit string Σ_i , which is used to represent solutions to a combinatorial optimization problem. For example, a 9-bit solution, 0-0-1-1-1-0-0-1-1 can be stored by the agent 8th as $\Sigma_8 = \{0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\}$. Each bit in the bit string is associated to a dimension as shown in Table 1. The length of the bit string is problem dependent and subjected to the size of the problem.

3.1. The Binary Simulated Kalman Filter

The first extension is called binary SKF (BSKF) [41]. In BSKF, modifications are needed only during initialization and generation of solution to combinatorial optimization problem. During the initialization of agents, a random bit string, Σ_i , is generated for each agent. During the generation of solution, the $K(t) \times |\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t+1)|$ term in (5) is mapped into a probabilistic value in the range of [0,1] using a mapping function, $S(\Delta i(t))$, as follows:

$$S(\Delta_i(t)) = |tanh(\Delta_i(t))| \tag{7}$$

where $\Delta_i(t) = K(t) \times |\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t+1)|$. Then, the probabilistic value is compared to a random number, $rand \in [0,1]$, to update the solution Σ_i . If the random number is smaller than the probabilistic value, the binary number at that particular dimension is complemented. For example, let $\Sigma_8\ (t) = \{0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\}$. If the binary bit in dimension 3 and 7 are subjected to complement, then the solution is updated to $\Sigma_8\ (t+1) = \{0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1\}$.

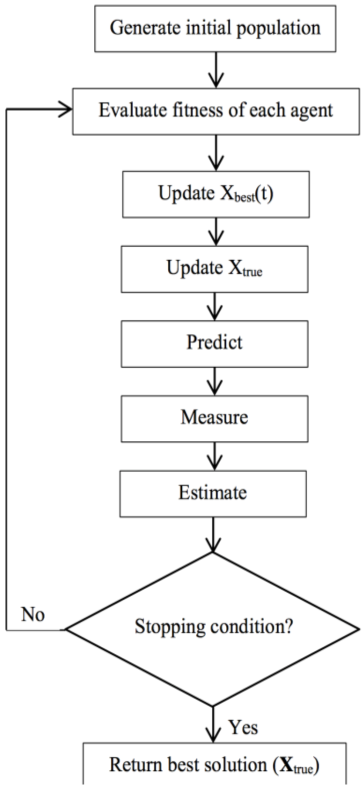
Generate initial population
Evaluate fitness of each agent
Update X_best(t)
Update X_true
Predict
Measure
Estimate
Stopping condition?
No
Yes
Return best solution (X_true)

Fig. 1: The original simulated Kalman filter algorithm [13-14].

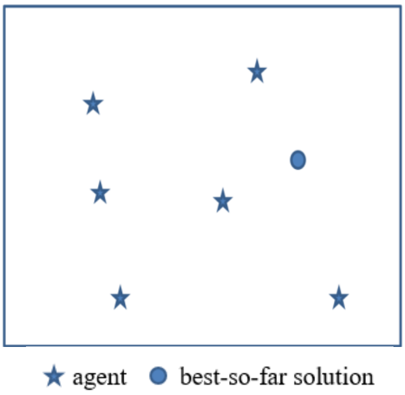

Fig. 2: Illustration of position of agents at the beginning of a search process [42-43].

Hence, the following rule can be used to update a bit value in the solution, Σ_i^d, for a particular dimension d:

```
if    $rand < S(\Delta i(t))$
then          $\Sigma_i^d(t+1) \leftarrow$ complement $\Sigma_i^d(t)$
else          $\Sigma_i^d(t+1) \leftarrow \Sigma_i^d(t)$
end
```

3.2. The Distance Evaluated Simulated Kalman Filter

In population-based search algorithm, generally, agents are randomly positioned in the search space. Then, the agents move in the search space to find global minimum or maximum. During the beginning of the search, exploration is preferred to make sure the search covers almost all regions in the search space. In this stage of search process, the position between agents is normally far with each other. As the search process continues, during the end of the search, exploration is no longer preferred because fine-tuning or exploitation is more preferred. During exploitation, agents becomes closer to each other and hence, the distance among them decreases.

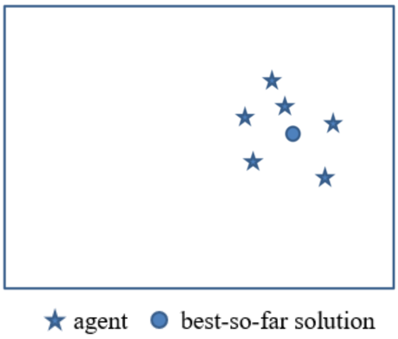

Fig. 3: Illustration of position of agents during the middle of a search process [42-43].

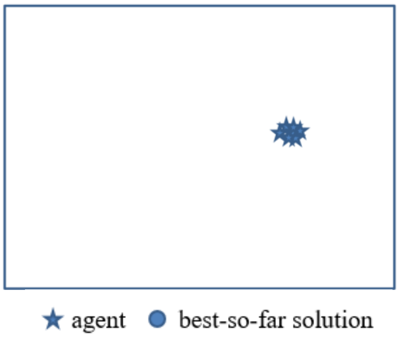

Fig. 4: Illustration of position of agents at the end of a search process [42-43].

The position of agents in a search space during a typical search process is illustrated in Figure 2, Figure 3, and Figure 4. Normally, as the iteration continues, the distance between agents and the best-so-far solution decreases. This distance plays an important role in the distance evaluated simulated Kalman filter algorithm (DESKF) [42-43]. A distance for a dimension d can be calculated as follows:

$$D_i^d(t) = x_i^d(t) - x_{true}^d(t) \tag{8}$$

The DESKF follows the same procedures as the BSKF. Instead of using the $K(t) \times |\boldsymbol{Z}_i(t) - \boldsymbol{X}_i(t|t+1)|$ term, the distance, $D_i^d(t)$, is mapped into a probabilistic value in the range of [0,1] as follows:

$$S\left(D_i^d(t)\right) = \left|tanhD_i^d(t)\right| \tag{9}$$

Then, the probabilistic value is compared to a random number, $rand \in [0, 1]$, to update the solution Σ_i . For a particular dimension d, a bit value in the solution, Σ_i^d is updated according to the same rule used in BSKF.

3.3. The Angle Modulated Simulated Kalman Filter

The angle modulated approach has been initially introduced for PSO [54]. The angle modulated simulated Kalman filter (AMSKF) combines the same approach with SKF to solve combinatorial optimization problems. This algorithm is called angle modulated SKF (AMSKF) [44] and its flowchart is shown in Figure 5. The main idea of the angle modulated approach in solving combinatorial optimization problem is to use a function, $g(x)$, to create a continuous signal. The shape of signal $g(x)$ is determined by 4 variables, namely, a, b, c, and d, using (10).

$$g(x) = \sin(2\pi(x-a) \times b \times \cos(A)) + d \tag{10}$$

where A = $2\pi(x-a) \times c$.

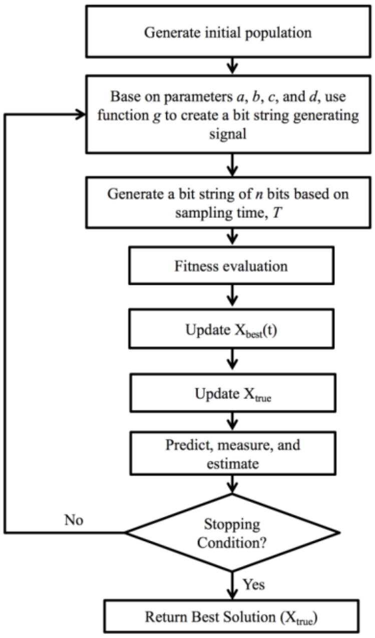
Generate initial population
Base on parameters a, b, c, and d, use function g to create a bit string generating signal
Generate a bit string of n bits based on sampling time, T
Fitness evaluation
Update $X_{best}(t)$
Update X_{true}
Predict, measure, and estimate
Stopping Condition?
No
Yes
Return Best Solution (X_{true})

Fig. 5: The angle modulated simulated Kalman filter algorithm [44].

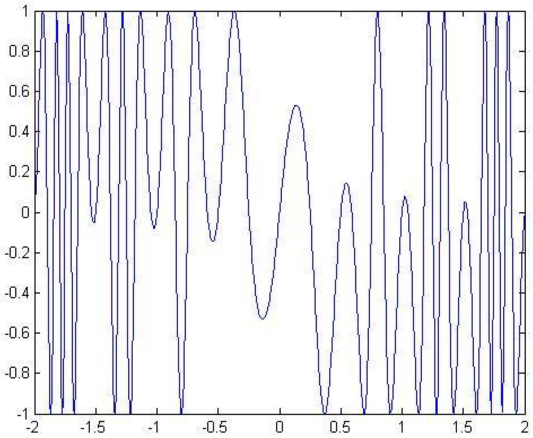

Fig. 6: An example of $g(x)$ plot [44].

Figure 6 shows an example of $g(x)$ plot for the case of a = 0, b = 1, c = 1, and d = 0. The region $g(x)$ > 0 is called binary 1 region and region $g(x)$ < 0 is called binary 0 region. After that sampling based on sampling time, T, is executed to generate a bit string of length n. The main advantage of angle modulated approach is that complex calculation in producing high dimensional bit string can be avoided. The search process in solving a combinatorial optimization problem can be done by tuning the values of a, b, c, and d only. In this work, the tuning is done by the SKF algorithm.

4. The Proposed State-Encoded Distance Evaluated Simulated Kalman Filter

Based on the state-encoded representation, a solution to a combinatorial optimization problem can be represented using state encoding. For example, let a solution 8-5-3-1-6-9-2-4-7 which is stored by the agent 4th as $\Sigma_4 = \{8\ 5\ 3\ 1\ 6\ 9\ 2\ 4\ 7\}$. Each state, from left to right, is associated to a dimension as shown in Table 2. The length of the bit string is problem dependent and subjected to the size of the problem.

The proposed state-encoded distance evaluated simulated Kalman filter (SEDESKF) requires the calculation of distance but unlike the existing DESKF, after all the distances are calculated, the distance is normalized as follows:

Table 2: Dimension number and state associate to the state for an instance of $\Sigma_4 = \{8\ 5\ 3\ 1\ 6\ 9\ 2\ 4\ 7\}$

Dimension number	State
1	8
2	5
3	3
4	1
5	6
6	9
7	2
8	4
9	7

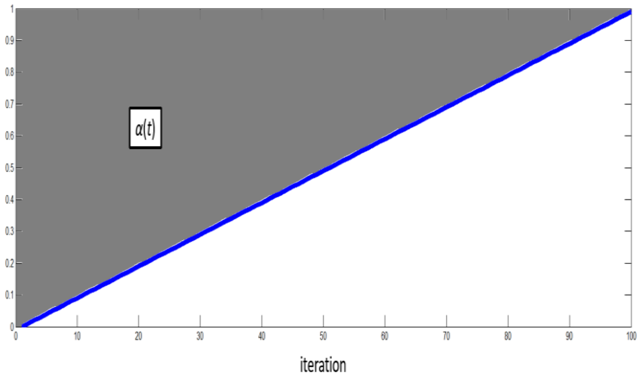

Fig. 7: The $\frac{t-1}{t_{max}}$ plot. The grey area indicates the values lie between $\frac{t-1}{t_{max}}$ and 1.

$$N_i^d(t) = [x_i^d(t) - x_{true}^d(t)] \times {}^{1}/_{distance_{max}^d(t)} \quad (11)$$

where $distance_{\max}^d(t)$ denotes the maximum distance value in dimension d at iteration t. Since the distance is normalized, the distance values are always in between 0 and 1, regardless of the iteration number. Hence, the impact of premature convergence of the SKF algorithm can be diminished. This normalized distance, $N_i^d(t) \in [0, 1]$, is used to update the solution in Σ_i using the following rule:

```
if    $\alpha(t) < N_i^d(t)$
then              $\Sigma_i^d(t+1) \leftarrow$ random state
else              $\Sigma_i^d(t+1) \leftarrow \Sigma_i^d(t)$
end
```

Using this rule, the next state depends on the variables $\alpha(t)$ and $N_i^d(t)$. If the $\alpha(t)$ value is greater than the $N_i^d(t)$ value, the current state is assigned as the next state. On the other hand, if the $N_i^d(t)$ value is greater than the $\alpha(t)$ value, a random state is assigned as the next state. The $\alpha(t) \in \left[\frac{t-1}{t_{max}}, 1\right]$ is a random number and its value is taken between $\frac{t-1}{t_{max}}$ (the lower limit) to 1 (the upper limit). The $\alpha(t)$ value is formulated to control the search behaviour in SEDESKF (to promote exploration during the early search process and to promote exploitation towards the end of the search process) and can be calculated as follows:

$$\alpha(t) = \left(1 - \left(\frac{t-1}{t_{max}}\right)\right) \times r + \frac{t-1}{t_{max}} \quad (12)$$

where $r \in [0, 1]$ is a random number.

To explain how the exploration and exploitation can be controlled during a search process, a linear graph $\frac{t-1}{t_{max}}$ is plotted and shown in Figure 7. The x-axis is t and t_{max}= 100. The grey area indicates the potential $\alpha(t)$ values at every iteration.

Let t =10. The lower bound and upper bound of $\alpha(t)$ are 0.09 and 1, respectively. The $\alpha(t)$ value calculated using (10) has higher probability to be lower than the $N_i^d(t) \in [0, 1]$. Hence, the state has more chance to change to another state.

Table 3: An instance that shows 4 dimensions (the 2nd, 5th, 8th, and 9th dimension) are subjected to state replacement

Dimension number	State at *t*	State at *t* + 1
1	8	8
2	5	9
3	3	3
4	1	1
5	6	2
6	9	9
7	2	2
8	4	8
9	7	4

Table 4: An instance that shows only 1 dimension (the 4th dimension) is subjected to state replacement

Dimension number	State at *t*	State at *t* + 1
1	8	8
2	5	5
3	3	3
4	1	9
5	6	6
6	9	9
7	2	2
8	4	4
9	7	7

Table 5: The SKF setting parameter

Parameter	Value
Iteration	1000
Number of agents	30
Number of trials	50
P(0)	1000
Q	0.5
R	0.5

Consider $t = 90$. Then, the lower bound and upper bound of $\alpha(t)$ are 0.89 and 1, respectively. In this case, the $\alpha(t)$ value calculated using (10) has higher probability to be greater than the $N_i^d(t) \in [0, 1]$. Hence, the state has less chance to change to another state. From the context of combinatorial optimization problem using state encoding, it is desirable to change the state during the early stage of the search process to explore more potential solutions. Hence, more state replacement is desirable during exploration. Following the example in Table 2, in which the solution stored by the agent 4th is $\Sigma_4\ (t) = \{8\ 5\ 3\ 1\ 6\ 9\ 2\ 4\ 7\}$. When more states are expected to change, the updated solution could be $\Sigma_4\ (t+1) = \{8\ 9\ 3\ 1\ 2\ 9\ 2\ 8\ 4\}$, as shown in Table 3. In this example, 4 dimensions are subjected to state changes. Specifically, the states at dimension 2, 5, 8, and 9 are replaced with new states.

As oppose to exploration, less state replacement is desired during exploitation phase. Using the same $\Sigma_4\ (t) = \{8\ 5\ 3\ 1\ 6\ 9\ 2\ 4\ 7\}$, as an example, as shown in Table 4, $\Sigma_4\ (t+1) = \{8\ 9\ 3\ 9\ 6\ 9\ 2\ 4\ 7\}$ since only one dimension is subjected to state replacement.

In this study, the performance of the proposed SEDESKF algorithm is evaluated by solving a set of travelling salesman problems (TSP). The objective of the TSP is to find the shortest distance from a start city to an end city while visiting every city not more than once. During the generation of TSP solution, repetition of states might exist which make a solution invalid. In this work, the same procedure to repair an invalid solution, which has been used in [45, 47, 49, 50], was employed. Figure 8 shows the flowchart of the proposed SEDESKF algorithm.

Table 6: The performance (in average)

TSP problem	SEDESKF	AMSKF	DESKF	BSKF
Berlin52	**22406.98**	22874.86	22932.2	22847.64
Bier127	**536858.9**	544059.5	544106.7	542440
Ch130	39426.15	39357.7	**39254.37**	39267
Ch150	**46136.63**	46168.05	46270.79	46174.03
D198	**143879.8**	158018.6	157618.5	158476.9
D493	**405901.5**	411931.2	411998.9	411621
D657	**794117**	796174.9	796175.3	796929.4
D1291	**1640368**	1646428	1645013	1648227
DSJ1000	**5.2E+08**	5.23E+08	5.24E+08	5.24E+08
Eil51	1267.804	**1266.809**	1268.42	2127.613
Eil76	2043.157	**2039.967**	2052.86	23782.28
Eil101	**2840.658**	2856.43	2845.66	2853.754
GIL262	**23718.7**	23851.59	23846.46	23853.9
KROA100	**135675.3**	136954.9	137043	137188.7
KROA150	**214278.5**	215813.7	216442.1	215796.9
KROA200	**289059.9**	291098.8	291940.4	291063.8
KROB100	**133147.7**	134818.2	134923.4	134786.5
KROB200	**283920**	285558.9	285802.7	286095.5
KROC100	**133605.2**	135858.8	135469.5	135539.3
KROD100	**130181.9**	131561.2	131622.3	131396.8
KROE100	**136381.9**	137716.4	138503.9	138610.7
LIN105	**98295.8**	98766.64	99036.19	99045.13
LIN318	**527431.1**	528817.1	**527049.5**	529112.7
P654	**1835950**	1848103	1845492	1849637
PCB442	**706661.8**	707728.3	708486.4	708016.9
PCB1173	**1332669**	1335124	1333055	1335923
PR76	**459240.7**	461176	461023.3	461949.7
PR107	**438474.9**	446571.5	446386.8	449263.3
PR124	**572756.4**	573148.5	580257.8	579691.2
PR136	**689707.8**	689959.7	690108.3	689880.4
PR144	**679453.9**	686191.1	682605.3	682410.8
PR152	**880010**	886369	886217.2	886457.3
PR226	**1472488**	1477167	1479082	1482490
PR264	**945954.9**	954069.8	954199	958776.8
PR299	**663592.6**	667263.2	664537	666494.6
PR439	**1714496**	1732577	1737005	1731523
PR1002	**6070399**	6078577	6085013	6079543
RAT99	**6632.39**	6718.733	6696.176	6732.771
RAT195	**19329.74**	19441.65	19422.96	19461.39
RAT575	**103555.9**	104311.9	103909.3	104248
RAT783	**165840**	167018.8	166512.9	166983
RD100	**45651.72**	45664.31	46096.3	45944.33
RL1304	**8880536**	8908134	8917743	8916299
RL1323	**9275698**	9303447	9303794	9302486
RL1889	**14114830**	14159546	14171974	14157634
ST70	2887.399	2902.092	**2882.996**	2890.875
TS225	1409269	1410333	1411955	**1409169**

5. Experiment, Result, and Discussion

Experimental setting parameters for SKF are shown in Table 5. The *P*, *Q*, and *R* values were chosen based on the previously published work on SKF [14]. In this paper, 47 instances of TSP are considered, from the size of 51 cities to 1400 cities. These problems were taken from TSPLib [55]. The averaged results are shown in Table 6. The values written in bold indicate the best result found for every case study.

The results shown in Table 6 indicate that the proposed SEDESKF outperforms the AMSKF, BSKF, and DESKF in 41 out of 47 case studies. At the first glance, it can be observed that SEDESKF exhibits a far superior performance in comparison to DESKF, AMSKF, and BSKF. The DESKF found three lowest distances, another two by AMSKF and one by BSKF.

A Friedman test was performed using the results in Table 6. As shown in Table 7, SEDESKF is ranked the best followed by AMSKF, DESKF, and BSKF. The Friedman statistic which is distributed according to chi-square with 3 degrees of freedom is equivalent to 68.131915. This shows significant different exist between SEDESKF and the benchmark algorithms.

Table 7: Result of the Friedman test

Algorithm	Ranking
SEDESKF	1.1702
AMSKF	2.766
DESKF	2.9574
BSKF	3.1064

Table 8: The result of the Holm post hoc analysis

i	Algorithms	z = (R0 − Ri)/SE	p	Holm
6	SEDESKF vs. BSKF	7.270313	0	0.008333
5	SEDESKF vs. DESKF	6.711058	0	0.01
4	SEDESKF vs. AMSKF	5.992016	0	0.0125
3	AMSKF vs. BSKF	1.278297	0.201145	0.016667
2	AMSKF vs. DESKF	0.719042	0.472115	0.025
1	DESKF vs. BSKF	0.559255	0.575988	0.05

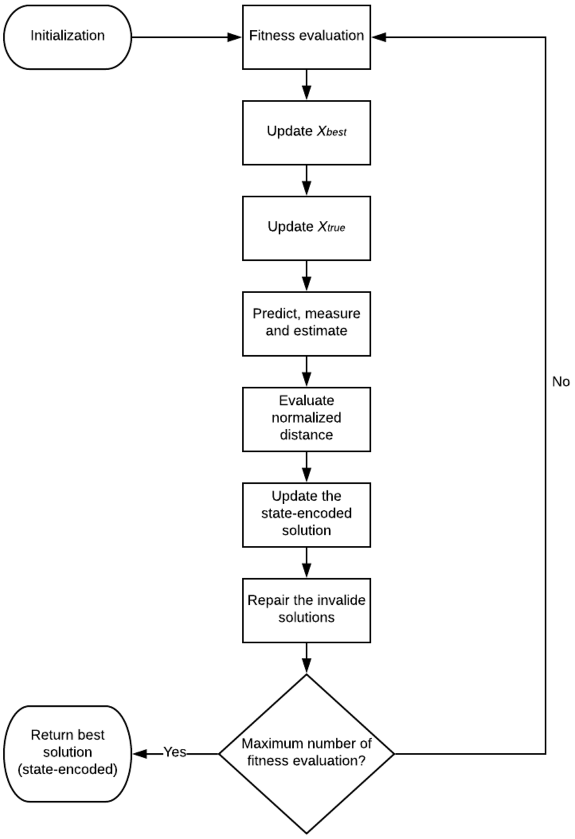
Initialization
Fitness evaluation
Update X_{best}
Update X_{true}
Predict, measure and estimate
Evaluate normalized distance
Update the state-encoded solution
Repair the invalide solutions
Maximum number of fitness evaluation?
Yes
No
Return best solution (state-encoded)

Fig. 8: The flowchart of the proposed SEDESKF algorithm.

A further analysis is performed using Holm post hoc procedure and its statistical value is tabulated in Table 8. The results of the Holm procedure show that SEDESKF is significantly better than the other algorithms with significance level, $\alpha = 0.05$.

6. Conclusion

Some problems such as TSP requires search space in binary or state. These problems are called combinatorial optimization problems. In computational intelligence, many optimization algorithms were extended for solving combinatorial optimization problems. A new extension of SKF is proposed in this paper. While the main concept of DESKF, which uses the calculated distance value, is adapted, the proposed SEDESKF employed the normalized distance value to control the exploitation and exploitation of the search. Specifically, the normalized distance is compared with a random number which has unity upper limit and dynamically increase lower limit. The experimental results showed that the proposed SEDESKF able to find the shortest path of most of the TSPs and in overall, significantly better than the existing BSKF, AMSKF, and DESKF algorithms.

Acknowledgement

This research is financially supported by an internal research fund granted by the Universiti Malaysia Pahang (UMP) (RDU1703133).

References

[1] Rani MR, Selamat H, Zamzuri H & Ibrahim Z (2012), Multi-objective Optimization for PID Controller Tuning using the Global Ranking Genetic Algorithm, *International Journal of Innovative Computing, Information and Control*, Vol. 8, Issue 1, pp. 269-284.

[2] Leung FHF, Lam HK, Ling SH & Tam PKS (2003), Tuning of the Structure and Parameters of a Neural Network using an Improved Genetic Algorithm, *IEEE Transaction on Neural Networks*, Vol. 14, Issue 1, pp. 79-88.

[3] Roshanaei M, Lucas C & Mehrabian AR (2009), Adaptive Beamforming using a Novel Numerical Optimisation Algorithm, *IET Microwaves, Antennas & Propagation*, Vol. 3, Issue 5, pp. 765-773.

[4] Li YL, Shao W, You L & Wang BZ (2013), An Improved PSO Algorithm and Its Application to UWB Antenna Design, *IEEE Antennas and Wireless Propagation Letters*, Vol. 12, pp. 1236-1239.

[5] Valenzuela CL & Jones AJ (1993), Evolutionary Divide and Conquer (I): A Novel Genetic Approach to the TSP, *Evolutionary Computation*, Vol. 1, Issue 4, pp. 313-333.

[6] Pedraza G, Diaz M & Lombera H (2016), An Approach for Assembly Sequence Planning by Genetic Algorithms, *IEEE Latin America Transactions*, Vol. 14, Issue 5, pp. 2066-2071.

[7] Montoya, J, Rathinam S & Wood Z (2014), Multiobjective Departure Runway Scheduling using Dynamic Programming, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, Issue 1, pp. 399-413.

[8] Mooney E & Dargen R (1996), Large-scale classroom scheduling, *IIE* Transactions, Vol. 28, No. 5, pp. 369-378.

[9] Uttraphan C, Shaikh-Husin N & Khalil Hani M, An Optimization Algorithm for Simultaneous Routing and Buffer Insertion with Delay-power Constraints in VLSI Layout Design, *Fifteenth International Symposium on Quality Electronics Design*, (2014), pp. 357-364.

[10] Jaffel Z & Farah M, A Symbiotic Organisms Search Algorithm for Feature Selection in Satellite Image Classification, *2018 4th International Conference on Advanced Technologies for Signal and Image Processing*, (2018), pp. 1-5.

[11] Goldberg DE & Holland JH (1988), Genetic Algorithms and Machine Learning, *Machine Learning*, Vol. 3, Issue 2-3, pp. 95-99.

[12] Dorigo M, Maniezzo V & Colorni A (1996), Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, Issue 1, pp. 29-41.

[13] 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.

[14] Ibrahim Z, Abdul Aziz NH, Ab. Aziz NA, Razali R, Mohamad MS (2016), Simulated Kalman Filter: A Novel Estimation-Based Metaheuristic Optimization Algorithm, *Advance Science Letters*, Vol. 22, pp. 2941-2946.

[15] Kalman RE (2011), A New Approach to Linear Filtering and Prediction Problems, *ASME Journal of Basic Engineering*, Vol. 82, No. 1, pp. 35-45.

[16] Muhammad B, Mat Jusof MF, Shapiai MI, Adam A, Md Yusof Z, Mohd Azmi KZ, Abdul Aziz NH, Ibrahim Z & Mokhtar N, Feature Selection using Binary Simulated Kalman Filter for Peak Classification of EEG Signals, *2018 8th International Conference on Intelligent Systems, Modelling and Simulation*, (2018), pp. 1-6.

[17] Adam A, Ibrahim Z, Mokhtar N, Shapiai MI, Mubin M & Saad I (2016), Feature Selection using Angle Modulated Simulated Kalman Filter for Peak Classification of EEG Signals, *SpringerPlus*, Vol. 5, No. 1580.

[18] Lazarus K, Noordin NH, Mat Jusof MF, Ibrahim Z, & Abas KH (2017), Adaptive Beamforming Algorithm Based on a Simulated Kalman Filter, *International Journal of Simulation: Systems, Science and Technology*, Vol. 18, No. 4, pp. 10.1-10.5.

[19] Lazarus K, Noordin NH, Mohd Azmi KZ, Abd Aziz NH & Ibrahim Z, Adaptive Beamforming Algorithm based on Generalized Opposition-based Simulated Kalman Filter, *The National Conference for Postgraduate Research, Pekan, Pahang*, (2016), pp. 1-9.

[20] Lazarus K, Noordin NH, Ibrahim Z, Mat Jusof MF, Mohd Faudzi MA, Subari N, and Mohd Azmi KZ, An Opposition-Based Simulated Kalman Filter Algorithm for Adaptive Beamforming, *IEEE International Conference on Applied System Innovation*, (2017), pp. 91-94.

[21] Lazarus K, Noordin NH, Ibrahim Z & Abas KH, Adaptive Beamforming Algorithm based on Simulated Kalman Filter, *Asia Multi Conference on Modelling and Simulation*, (2016), pp. 19-23.

[22] Abd Aziz NH, Ab. Aziz NA, Ibrahim Z, Razali S, Abas KH, Mohamad MS, A Kalman Filter Approach to PCB Drill Path Optimization Problem, *IEEE Conference on Systems, Process and Control*, (2016), pp. 33-36.

[23] Abdul Aziz NH, Ibrahim Z, Ab. Aziz NA, Md Yusof Z & Mohamad MS (2018), Single-Solution Simulated Kalman Filter Algorithm for Routing in Printed Circuit Board Drilling Process, *Intelligent Manufacturing and Mechatronics*, pp. 649-655.

[24] Mustapa A, Md Yusof Z, Adam A, Muhammad B & Ibrahim Z (2018), Solving Assembly Sequence Planning using Angle Modulated Simulated Kalman Filter, *IOP Conference Series: Materials, Science, and Engineering*, Vol. 319, No. 012044.

[25] Md Yusof Z, Satiman SN, Mohd Azmi KZ Badaruddin Muhammad, Saifudin Razali, Zuwairie Ibrahim, Zulfakar Aspar, and Suraya Ismail, Solving Airport Gate Allocation Problem using Simulated Kalman Filter, *International Conference on Knowledge Transfer*, (2015), pp. 121-127.

[26] Mohd Azmi KZ, Md Yusof Z, Satiman SN, Muhammad B, Razali S, Ibrahim Z, Ab. Aziz NA & Abd Aziz NH, Solving Airport Gate Allocation Problem Using Angle Modulated Simulated Kalman Filter, *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 875-885.

[27] Ann NQ, Pebrianti D, Bayuaji L, Daud MR, Samad R, Ibrahim Z, Hamid R & Syafrullah M (2018), SKF-Based Image Template Matching for Distance Measurement by using Stereo Vision, *Intelligent Manufacturing and Mechatronics*, pp. 439-447.

[28] Ann NQ, Pebrianti D, Ibrahim Z, Mat Jusof MF, Bayuaji L, and Abdullah NRH (2018), Illumination-Invariant Image Matching Based on Simulated Kalman Filter (SKF), *Journal of Telecommunication, Electronics and Computer Engineering*, Vol. 10, No. 1-3, pp. 31-36.

[29] Muhammad B, Mohd Azmi KZ, Ibrahim Z, Mohd Faudzi AA & Pebrianti D, Simultaneous Computation of Model Order and Parameter Estimation for System Identification Based on Opposition-Based Simulated Kalman Filter, *SICE International Symposium on Control Systems 2018, March 9-11, Tokyo, Japan*, (2018), pp. 105-112.

[30] Mohd Azmi KZ, Ibrahim Z, Pebrianti D & Mohamad MS (2017), Simultaneous Computation of Model Order and Parameter Estimation for ARX Model Based on Single and Multi Swarm Simulated Kalman Filter, *Journal of Telecommunication, Electronic, and Computer Engineering*, Vol. 9, No. 1-3, pp. 151-155.

[31] Muhammad B, Pebrianti D, Abdul Ghani N, Abdul Aziz NH, Ab. Aziz NA, Mohamad MS, Shapiai MI & Ibrahim Z, An Application of Simulated Kalman Filter Optimization Algorithm for Parameter Tuning in Proportional-Integral-Derivative Controllers for Automatic Voltage Regulator System, *SICE International Symposium on Control Systems 2018, March 9-11, Tokyo, Japan*, (2018), pp. 113-120.

[32] Abd Aziz NH, Ibrahim Z, Razali S, Bakare TA & Ab. Aziz NA, How Important the Error Covariance in Simulated Kalman Filter?, *The National Conference for Postgraduate Research, Pekan, Pahang*, (2016), pp. 315-320.

[33] Abdul Aziz NH, Ab. Aziz NA, Mat Jusof MF, Razali S, Ibrahim Z, Adam A & Shapiai MI, An analysis on the Number of Agents Towards the Performance of the Simulated Kalman Filter Optimizer, *2018 8th International Conference on Intelligent Systems, Modelling and Simulation*, (2018), pp. 16-21.

[34] Abdul Aziz NH, Ibrahim Z, Ab. Aziz NA & Razali S (2017), Parameter-less Simulated Kalman Filter, *International Journal of Software Engineering and Computer Systems*, Vol. 3, pp. 129-137.

[35] Abd Aziz NH, Ab. Aziz NA, Ibrahim Z, Razali S, Mat Jusof MF, Abas KH, Mohamad MS & Mokhtar N, Simulated Kalman Filter with Randomized Q and R Parameters, *International Conference on Artificial Life and Robotics*, (2017) pp. 711-714.

[36] Badaruddin Muhammad, Zuwairie Ibrahim, Mohd Falfazli Mat Jusof, Nor Azlina Ab Aziz, Nor Hidayati Abd Aziz, and Norrima Mokhtar, A Hybrid Simulated Kalman Filter - Gravitational Search Algorithm (SKF-GSA), *International Conference on Artificial Life and Robotics*, (2017), pp. 707-710.

[37] Muhammad B, Ibrahim I, Mohd Azmi KZ, Abas KH, Ab. Aziz NA, Abd Aziz NH & Mohamad MS, Performance Evaluation of Hybrid SKF Algorithms: Hybrid SKF-PSO and Hybrid SKF-GSA, *The National Conference for Postgraduate Research, Pekan, Pahang, Malaysia*, (2016), pp. 865-874.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Ibrahim I, Md Yusof Z, Nawawi SW, Rahim, MAA, Khalil K, Ahmad H & Ibrahim Z, A Novel Multi-state Particle Swarm Optimization for Discrete Combinatorial Optimization Problems, *Fourth International Conference on Computational Intelligence, Modelling and Simulation*, (2012), pp. 18-23.

[46] Ibrahim I, Ahmad H, Ibrahim Z, & Md Yusof Z (2015), An Improved Multi-State Particle Swarm Optimization for Discrete Combinatorial Optimization Problems, *International Journal of Simulation - Systems, Science and Technology*, Vol. 16, No. 6, pp. 14.1-14.8.

[47] Ibrahim I, Ahmad H, Ibrahim Z, Mat Jusoh MF, Zulkifli Md. Yusof, Nawawi SW, Khalil K & Abdul Rahim MA (2014), Multi-State Particle Swarm Optimization for Discrete Combinatorial Optimization Problem, *International Journal of Simulation - Systems, Science and Technology*, Vol. 15, No. 1, pp. 15-25.

[48] Ibrahim I, Ibrahim Z, Ahmad H & Md Yusof Z, An Improved Multi-State Particle Swarm Optimization for Discrete Optimization Problems, *The 7th International Conference on Computational Intelligence, Communication Systems, and Networks*, (2015), pp. 3-8.

[49] Ibrahim I, Ibrahim Z, Ahmad H & Md. Yusof Z, A Multi-State Gravitational Search Algorithm for Combinatorial Optimization Problems, *The 7th International Conference on Computational Intelligence, Communication Systems, and Networks,* (2015), pp. 9-14.

[50] Ibrahim I, Ahmad H, Ibrahim Z & Md Yusof Z (2015), A Novel Multi-State Gravitational Search Algorithm for Discrete Optimization Problems, *International Journal of Simulation - Systems, Science and Technology*, Vol. 16, No. 6, pp. 15.1-15.8.

[51] Ismail Ibrahim, Zuwairie Ibrahim, Hamzah Ahmad, and Zulkifli Md Yusof (2015), An Assembly Sequence Planning Approach with a Multi-State Gravitational Search Algorithm, *ARPN Journal of Engineering and Applied Sciences*, Vol. 10, No. 22, pp. 10709-10715.

[52] Ibrahim I, Ibrahim Z, Ahmad H, Mat Jusof MF, Md. Yusof Z, Nawawi SW & Marizan Mubin M (2015), An Assembly Sequence Planning Approach with a Rule-based Multi-State Gravitational Search Algorithm, *The International Journal of Advanced Manufacturing Technology*, Vol. 79, Issue 5, pp. 1363-1376.

[53] Ibrahim I, Ibrahim Z, Ahmad H & Md Yusof Z, Rule-Based Multi-State Gravitational Search Algorithm for Discrete Optimization Problem, *The 4th International Conference on Software Engineering and Computer Systems (ICSECS2015), Kuantan, Pahang*, (2015).

[54] Leonard BJ & Engelbrecht AP (2014), Angle Modulated Particle Variants, *Lecture Notes on Computer Science*, Vol. 8667, pp. 38-49.

[55] http://www.iwr.uni-heidelberg.de/groups/comopt/software