Global Optimum Distance Evaluated Particle Swarm Optimization for Combinatorial Optimization Problem

Ainizar Mustapa, Zuwairie Ibrahim, Zulkifli Md Yusof, Ismail Ibrahim
Faculty of Electrical and Electronics Engineering
Universiti Malaysia Pahang
26600 Pekan, Malaysia

Abstract—Based on the mechanism of Particle Swarm Optimization (PSO) measurement process, every particle estimates the global minimum/maximum. Particles communicate among them to update and improve the solution during the search process. However, the PSO is only capable to solve continuous numerical optimization problem. In order to solve discrete optimization problems, a new global optimum distance evaluated approach is proposed and combined with PSO. A set of traveling salesman problems (TSP) are used to evaluate the performance of the proposed global optimum distance evaluated PSO (GO-DEPSO). Based on the analysis of experimental results, we found that the proposed DEPSO is capable to solve discrete optimization problems using TSP.

Keywords—*Particle Swarm Optimization; Distance Evaluated; Combinatorial; Traveling Salesman Problems*

1. Introduction

Over the last decades there has been a growing interest in algorithms inspired by the observation of natural phenomenon. It has been shown by many researches that these algorithms are good replacement as tools to solve complex computational problems. Various heuristic approaches have been adopted by researches including genetic algorithm, tabu search, simulated annealing, ant colony and particle swarm optimization. PSO can be classified in swarm intelligence areas, where developed by Kennedy and Eberhart in 1995. Since 1995, PSO is being researched and utilized in different subjects by researches around the world. It is reported in the literature that the PSO technique can generate high-quality solution within shorter calculation time on some optimization problems.

PSO is motivated from the simulation of social behavior. This optimization approach update the population of individuals by applying an operator according to the fitness information obtained from the environment so that the individuals of the population can be expected to move towards better solution areas [1]. The PSO technique conducts searches using a population of particles, corresponding to individuals. Each particle tries to search the best position (state) with time in a multidimensional space and adjusts its position in light of its own experience and the experiences of its neighbors, including the current velocity and position and the best previous position experienced by itself and its neighbors [1].

The origin version of the particle swarm has been operated in continuous space. But many optimization problems are set in combinatorial/discrete space. There are a lot of discrete optimization problems in literature and real-world applications. Examples of discrete optimization problems are assembly sequence planning [2-3], DNA sequence design [4-5], VLSI routing [6-7], robotics drill route problem [8], and airport gate allocation problem [9].

This paper is organized as follows. At first, PSO will be briefly reviewed followed by a detail description of the proposed global optimum distance evaluated PSO (GO-DEPSO) algorithm. Experimental set up will be explained and results will be shown and discussed. Lastly, a conclusion will be provided at the end of this paper.

2. Particle Swarm Optimization

Similar to evolutionary algorithms, the PSO technique conducts searches using a population of particles, corresponding to individuals. Each particle represents a candidate solution to the problem at hand. In a PSO algorithm, particles change their positions by flying around in a multidimensional search space until a relatively unchanged position has been encountered, or until computational limitations are exceeded. The PSO algorithm is illustrated in Figure-1.

Bird flocking optimizes a certain objective function. Each particle (individual) knows its best value so far and its position (called as personal best or $pbest_i$ for i^{th} particle). The information corresponds to personal experiences of each particle. Moreover, each particle knows the best value so far in the group among (known as global best or *gbest*). Namely, each particle tries to modify its position x_i) using the following information:

- the distance between the current position and *pbest*.
- the distance between the current position and *gbest*.

This modification can be represented by the concept of velocity (v_i). The velocity and the position of each particle in a d -dimensional space, can be modified by the following equations [1]:

$$v_i^{k+1} = w.v_i^k + c_1.r_1.(pbest_i - s_i^k) + c_2.r_2.(gbest - s_i^k) \quad (1)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (2)$$

where r_1 and r_2 are two random functions in the range [0,1], c_1 and c_2 are cognitive and social coefficient accordingly and w is the inertia weight. $s_i = (s_i^1, s_i^2,..., s_i^{k+1})$ represents the position of i^{th} particle. The rate of the position change (velocity) for particle i is represented as $v_i = v_i^1, v_i^2,..., v_i^{k+1}$). The best previous position (the position giving the best fitness value) of the i^{th} particle is recorded and represented as $pbest_i = (pbest_{i1}, pbest_{i2},..., pbest_{id})$.

As a particle moves through the search space, it compares its fitness value at the current position to the best fitness value it has ever attained at any time up to the current time. There are two PSO models known as global model (or *gbest*) and personal model (or *pbest*). The equations (1) and (2) represent global model. In the global model, all the particles in the swarm interact with the *gbest* while in the personal model, each particle interact with the local best particle ($pbest_i$ for i^{th} particle).

3. GLOBAL OPTIMUM DISTANCE EVALUATED PARTICLE SWARM OPTIMIZATION

In population-based search algorithm, generally, particles are randomly positioned in the search space. Then, the particles 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 articles 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, particles becomes closer to each other and hence, the distance among them decreases.
The position of particles 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 particles and the best-so-far solution decreases. This distance plays an important role in the proposed global optimum distance evaluated particle swarm optimization algorithm (GO-DEPSO).

In GO-DEPSO, the distance is mapped into a probabilistic value [0,1] and then the probabilistic value will be compared with a random number [0,1] to update a bit string or solution to a combinatorial optimization problem. In detail, most of the calculations in the proposed GO-DEPSO are similar to the original PSO. Modifications are needed only during initialization and generation of solution to combinatorial optimization problem.

During the initialization of particles, in PSO, the states of each particle is given randomly. An additional initialization is introduced in GO-DEPSO. Every particle is associated with a random bit string as well. The length of the bit string is problem dependent and subjected to the size of the problem. Thus, 2 types of variables are associated with an particle in PSO. They are continuous variable, $\boldsymbol{x}$, which is produced as estimated value of PSO (also similar to the position of particle in a search space), and a bit string, $\boldsymbol{\Sigma}$, which is used to represent solution to a combinatorial optimization problem.

In GO-DEPSO, for a particular d^{th} dimension, the distance between an i^{th} agent to the best-so-far solution at iteration t can be calculated as follows:

$$D_i^k(t) = x_i^k(t) - x^k_{best\text{-}so\text{-}far}(t) \quad (3)$$

In binary gravitational search algorithm (BGSA) [12], a function, as shown in Figure-5, is used to map a velocity value into a probabilistic value within interval [0,1]. Similar function is used in GO-DEPSO. This distance value, $D_i^k(t)$, is mapped to a probabilistic value within interval [0,1] using a probability function, $S(D_i^k(t))$, as follows:

$$S(D_i^k(t)) = |\tanh(D_i^k(t))| \quad (4)$$

After the $S(D_i^k(t))$ is calculated, a random number, *rand*, is generated and a binary value at dimension d of an i^{th} agent, Σ_i^d , is updated according to the following rule:

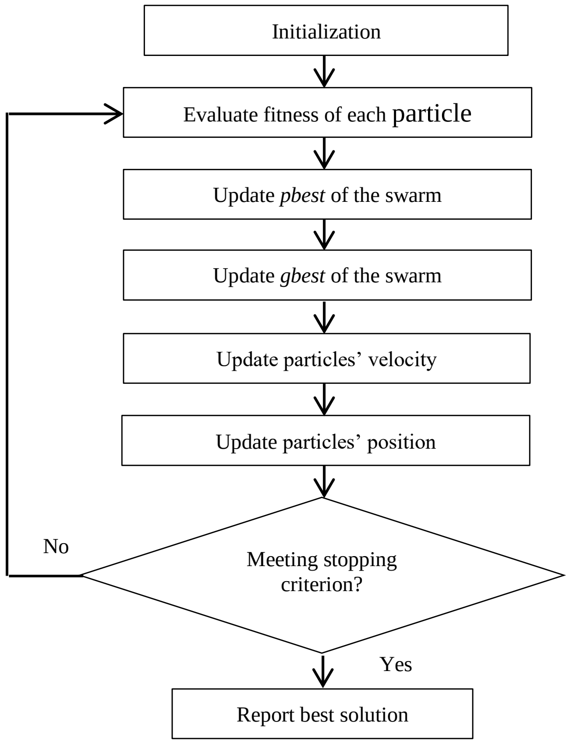

Figure 1: PSO algorithm

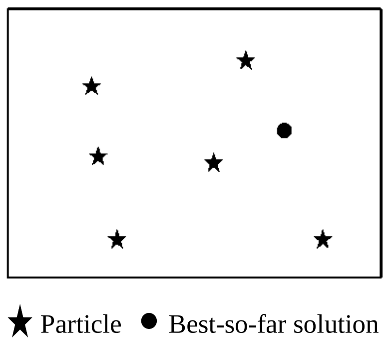

Figure 2: Position of particles at the beginning of the search.

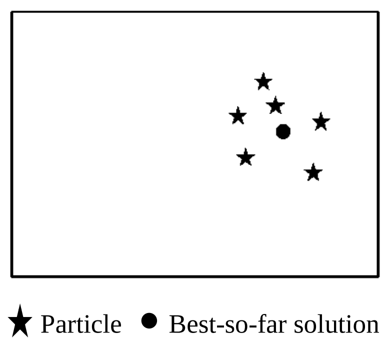

Figure 3: Position of particles during the middle of the search.

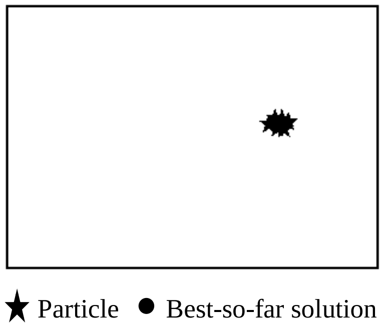

Figure 4: Position of particles at the end of the search.

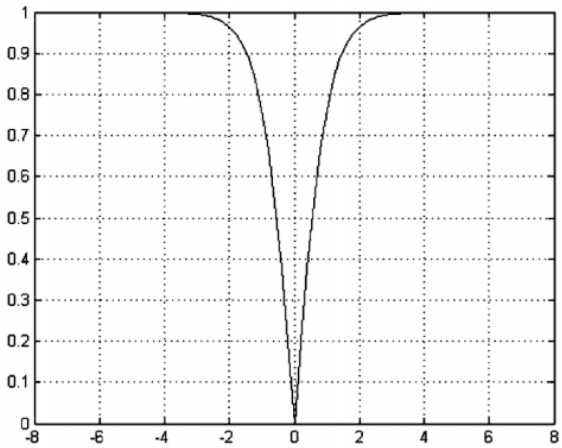

Figure 5: Probabilistic function used in [12]

$$\begin{gathered}\text{if rand} < S\,(D_i^{\,k}\,(t)\,) \\ \text{then } \Sigma_i^d\,(t+1) = \text{complement } \Sigma_i^d\,(t+1) \\ \text{else } \Sigma_i^d\,(t+1) = \Sigma_i^d\,(t+1)\end{gathered} \tag{5}$$

The GO-DEPSO algorithm is illustrated in Figure-6.

The GO-DEPSO is applied to solve a set of TSP. The objective of TSP is to find the shortest distance from a start city to an end city while visiting every city not more than once. In this paper, 10 instances of TSPs are considered, from the size of 52 cities to 2103 cities, as shown in Table-1. These problems were taken from TSPLib [10]. Experimental setting for GO-DEPSO is shown in Table-2. For benchmarking purpose, 1 additional experiment was considered, which is based on the well-established distance-evaluated simulated kalman filter (DESKF) [13]. Experimental setting for DESKF is shown in Table-3. In all experiments, the number of runs, the number of particles or agents, and the number of iterations are 50, 30, and 1000, respectively.

5. EXPERIMENTS, RESULTS AND DISCUSSION

The proposed GO-DEPSO is compared with DESKF. The average performances of the algorithms are presented in Table-4. The numbers written in bold show the best performance. Based on these average performances, Wilcoxon signed rank test is performed. The result of the test is tabulated in Table-5.

Based on these average performances, Wilcoxon signed rank test is performed. The result of the test is tabulated in Table-5. The level of significant chosen here is 0.05. It is found that statistically no significant difference is found between the roposed GO-DEPSO and DESKF. Both of the algorithms perform as good as each other in solving TSP problems. Examples of convergence curves of GO-DEPSO are shown in Figure-7 and Figure-8.

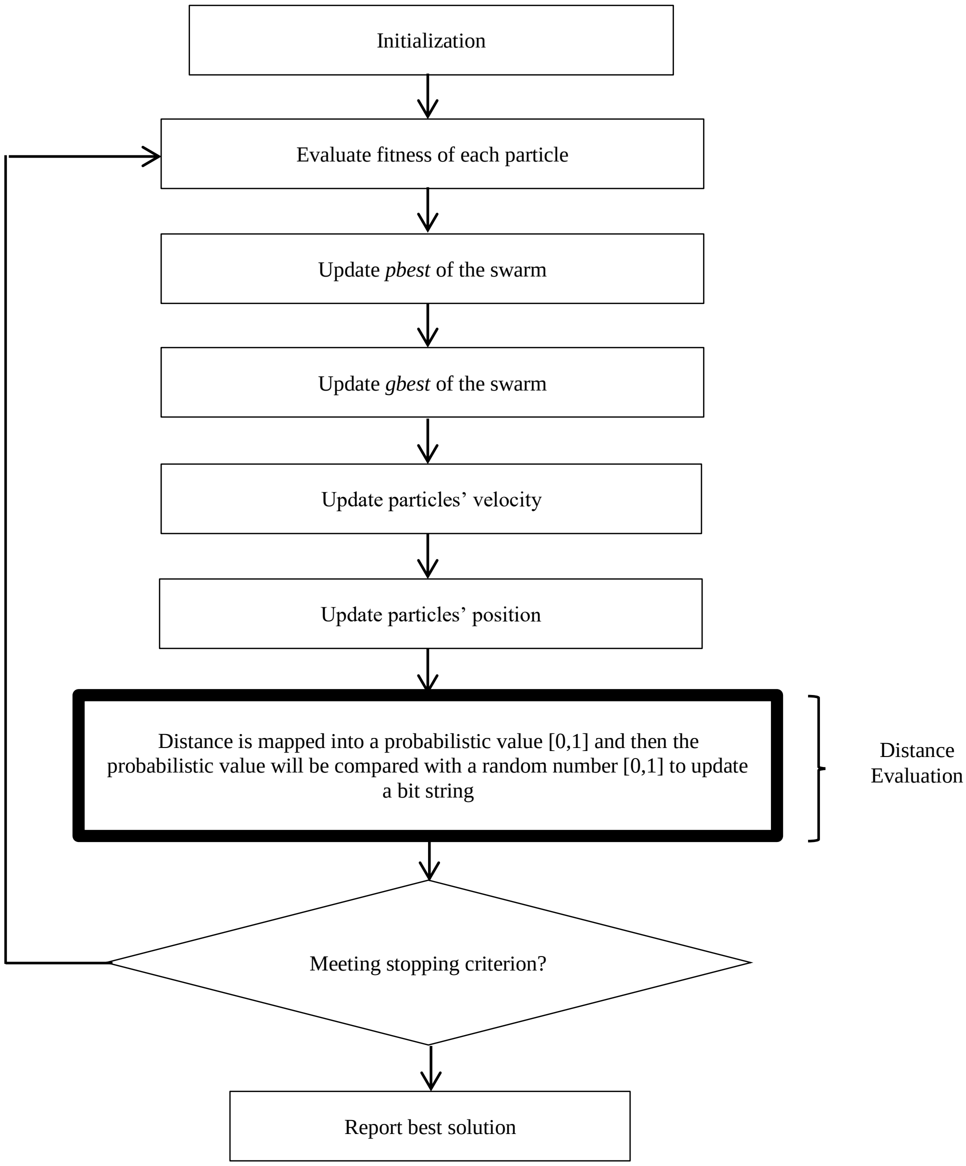

Figure 6: GO-DEPSO algorithm

6. Conclusion

This paper reports a research on using a modified PSO for solving combinatorial optimization problems. Based on the proposed GO-DEPSO, the distance between a particle to the best-so-far solution is evaluated to update a binary value. Experimental result and analysis showed the potential of GO-DEPSO. The GO-DEPSO performed as good as DESKF.

7. Reference

[1] H. Nezamabadi-pour, M. Rostami-shahrbabaki, M.M. Farsangi, ″Binary Particle Swarm Optimization: challenges and New Solutions″, The Journal of Computer Society of Iran (CSI) On Computer Science and Engineering (JCSE), vol. 6, no. (1-A), pp. 21-32, 2008.

[2] Mukred J.A.A, Ibrahim Z., Ibrahim I., Adam A., Wan K., Yusof Z.M., and Mokhtar N. 2012. ″A Binary Particle Swarm Optimization Approach to Optimize Assembly Sequence Planning″, Advance Science Letters, Vol. 13, pp. 732-738.

Table-1. Property of the test problems.

TSP Index	Name	Size
1	Berlin52	52
2	Bier127	127
3	Ch130	130
4	Ch150	150
5	D198	198
6	D493	493
7	D657	657
8	D1291	1291
9	D2103	2103
10	DSJ1000	1000

Table-2. Experimetal setting parameters GO-DEPSO.

Parameters	Value
rand	[0,1]
ϖ	0.9 - 0.4
v	-4 - 4
c_1	2
c_2	2

Table-3. Experimetal setting parameters DESKF.

Parameters	Value
P	1000
Q	0.5
R	0.5
rand	[0,1]
x_{min}	-100
x_{max}	100

Table-4. Average performance.

TSP Index	GO-DEPSO	DESKF
1 Berlin52	**22903.03**	22932.2
2 Bier127	545544.72	**544106.72**
3 Ch130	39290.75	**39254.37**
4 Ch150	**46187.73**	46270.79
5 D198	**157253.68**	157618.45
6 D493	412033.06	**411998.9**
7 D657	**795856.84**	796175.26
8 D1291	**1643490.5**	1645013.36
9 D2103	3124524.54	**3123370**
10 DSJ1000	**523998219**	524027900

Table-5. Wilcoxon test result.

Comparison	R^+	R^-
GO-DEPSO vs DESKF	21	34

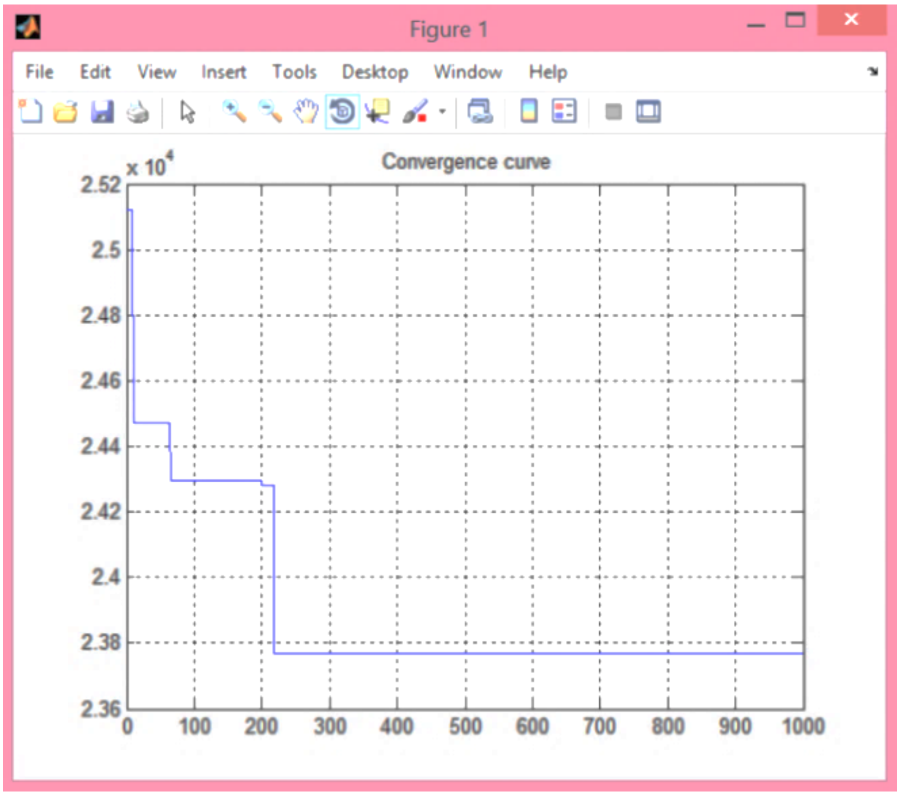

Figure-7. An example of convergence curve of GO-DEPSO for TSP index 1. Note that y-axis is the fitness value and x-axis is the iteration value.

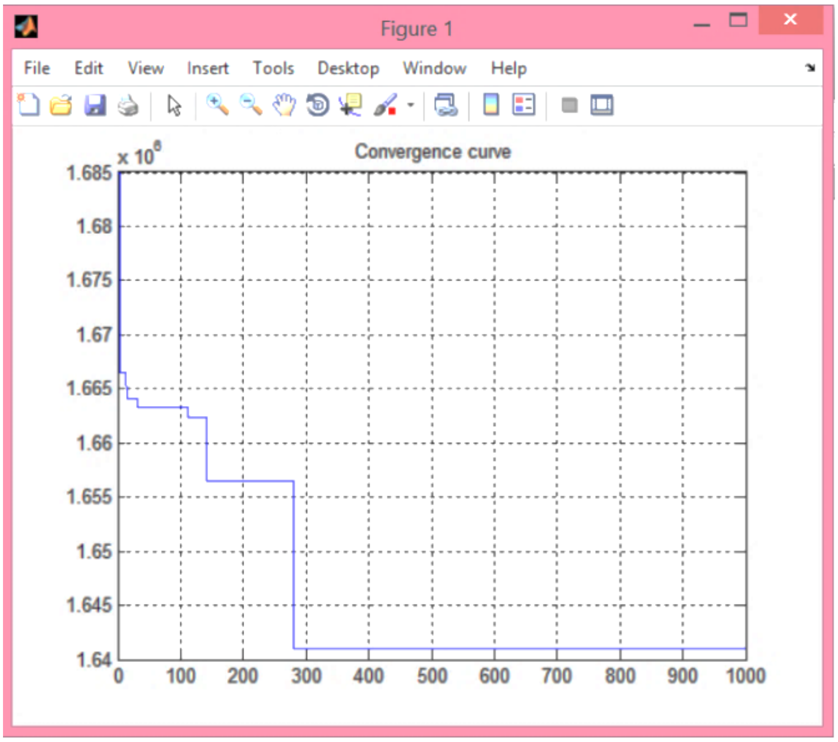

Figure-8. An example of convergence curve of GO-DEPSO for TSP index 8.
Note that y-axis is the fitness value and x-axis is the iteration value.

[3] Ibrahim I., Ibrahim Z., Ahmad H., Mat Jusof M.F, Yusof Z.M., Nawawi S.W., and 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, pp. 1363-1376.

[4] Yakop F., Ibrahim Z., Zainal Abidin A.F., Yusof Z.M., Mohamad M.S., Wan K., and Watada J. 2012. ″An Ant Colony System for Solving DNA Sequence Design Problem in DNA Computing″, International of Innovative Computing, Information and Control, Vol. 8, No. 10, pp. 7329-7339.

[5] Ibrahim Z., Khalid N.K., Ibrahim I., Lim K.S., Bunyamin S., Yusof Z.M., and Muhammad M.S. 2011. ″Function Minimization in DNA Sequence Design Based on Binary Particle Swarm Optimization″. Journal Teknologi D (UTM), No. 54, pp. 331-342.

[6] Yusof Z.M., Zainal Abidin A.F., Salam M.N.A., Khalil K., Mukred J.A.A., Hani M.K., and Ibrahim Z. 2011. ″A Binary Particle Swarm Optimization Approach for Buffer Insertion in VLSI Routing″. International Journal of Innovative Management, Information and Production, Vol. 2, No. 3, pp. 34-39.

[7] Yusof. Z.M., Zainal Abidin A.F., Adam A., Khalil K., Mukred J.A.A., Mohamad M.S., Hani M.K., and Ibrahim Z. 2012. ″A Two Step Binary Binary Particle Swarm Optimization Approach for Routing in VLSI″. ICIC Express Letters, Vol. 6, No. 3, pp.771-776.

[8] Othman M.H., Zainal Abidin A.F., Adam A., Yusof Z.M., Ibrahim Z., Mustaza S.M., and Lai Y.Y. 2011. ″A Binary Particle Swarm Optimization Approach for Routing in PCB Holes Drilling Process″, International Conference on Robotic Automation System (ICORAS2011), pp. 201-206.

[9] Bouras A., Ghaleb M.A., Suryahatmaja U.S., and Salem A.M. 2014. ″The Airport Gate Assignment Problem: A Survey″. The Scientific World Journal.

[10] comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[11] Kennedy, J. and Eberhart, R. 1997. ″A discrete binary version of the particle swarm algorithm″. IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, pp. 4104–4108.

[12] Rashedi, E., Nezamabadi-pour, H., and Saryazdi. S. 2010. ″BGSA: binary gravitational search algorithm″. Natural Computing, vol. 9, issue 3, pp. 727-745.

[13] Yusof, ZM, Ibrahim, Z, Ibrahim, I, Azmi, KZM, Aziz, NAA, Aziz, NHA, Mohamad, MS. 2015. "Distance evaluated simulated kalman filter for combinatorial optimization problem". International Conference on Electrical Control and Computer Engineering 2015.