International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 13, Number 1 (2020), pp. 117-136
© International Research Publication House. https://dx.doi.org/10.37624/IJERT/13.1.2020.117-136

A Significant Feature Selection in the Mahalanobis Taguchi System using Modified-Bees Algorithm

Faizir Ramlie[1,] Wan Zuki Azman Wan Muhamad[2,*], Khairur Rijal Jamaludin[3], Elizabeth Cudney[4] and Rozzeta Dollah[5]

[1,3,5]Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia,
Jalan Sultan Yahya Petra, 54100 Kuala Lumpur, Malaysia
[2]Institute of Engineering Mathematics, Universiti Malaysia Perlis
Kampus Pauh Putra, 02600 Arau, Perlis, Malaysia
[4]Department of Engineering Management and System Engineering
Missouri University of Science and Technology: Rolla, MO, US

[1]ORCID: 0000-0003-1521-3415
[2]ORCID: 0000-0001-5697-8196
[3]ORCID: 0000-0001-9110-753X
[4]ORCID: 0000-0002-3990-0474
[5]ORCID: 0000-0002-6808-6491

Abstract

This paper compares the performance of orthogonal array (OA), modified-Bees Algorithm (*mBA*) and conventional Bees Algorithm (BA) in significant feature selection scheme (optimization) of the Mahalanobis-Taguchi System (MTS) methodology. The main contribution of this work is to address both performances in terms of computing cost i.e. computing time as well as classification accuracy rate. Several studies have been conducted to evaluate the performance of OA against other heuristic search techniques in MTS methodology however, discussions in terms of the computing speed performances were found to be lacking. Instead, the accuracy performances were given the emphasis by drawing criticisms towards the deployment of OA as ineffective as compared to other state-of-the-art heuristic algorithms. Bees Algorithm (BA) is one heuristic search technique that discovers optimal (or near optimal) solutions using search strategy mimics the social behaviour of a honeybee colony. In this comparison work, modified-BA (*mBA*) is introduced into the optimization scheme of MTS with a modification on its neighbourhood search mechanism from the original BA. Instead of searching in random mode, a backward selection method is proposed. MD is used as the result assessment metric while the larger-the-better type of SNR is deployed as the algorithm's objective function. The historical heart liver disease data are used as the case study on which the comparisons between OA, *m*BA and BA performances specifically in terms of the computing speed are made and addressed. The outcomes showed a promising performance of the *m*BA as compared to OA with a comparable classification accuracy rate. Eventhough OA outperformed *mBA* in terms of computational speed, the MTS manage to classify at the expense of lower number of variables suggested by *m*BA. The *m*BA also converges faster than the conventional BA in finding the potential solution of the case problem.

Keywords: Mahalanobis-Taguchi System, Orthogonal Array, Bees Algorithm, Feature Selection, Swarm Intelligence

1 INTRODUCTION

The Mahalanobis–Taguchi System (MTS) is a pattern information technology that aids quantitative decision making process by constructing a multivariate measurement scale using data analytic methods (Ramlie, Jamaludin, & Dolah, 2016). It was developed by the renowned Japanese Quality guru Dr. Genichi Taguchi. The MTS methodology started with the theory of Mahalanobis distance (MD) formulated by the famous Indian statistician, Dr. P.C. Mahalanobis in 1936 (Mahalanobis, 1936) inspired from his determination to examine if the Indian people who married European people came from specific caste levels. The formulation of MD was then extended by Dr. Taguchi whom integrated the MD formulation with his robust engineering concepts to enhance the MD methodology to become a popular application tool for pattern recognition and forecasting technique in multidimensional systems (Muhamad, Jamaludin, Yahya, & Ramlie, 2017). Therefore, numerous applications of MTS ranging from the field remanufacturing, medical diagnosis, pattern recognitions, aerospace, agro-cultures, administration, banking and finances have been reported (Abu, Jamaludin, & Ramlie, 2013; Ghasemi, Aaghaie, & Cudney, 2015; Muhamad, Ramlie, & Jamaludin, 2017; Muhamad, Jamaludin, Ramlie, Harudin, & Jaafar, 2017).

Despite the great acceptance by various academicians and industrial communities since its introduction by Taguchi, one operational aspect of MTS related to the deployment of orthogonal array (OA) as the feature selection algorithm in MTS has been criticised by many (Hawkins, 2003, Woodall et al., 2003, Abraham & Variyath, 2003, Pal & Maiti, 2010). Hawkins (2003) argued that OA is a type of fractional factorial design of experiment scheme, thus it is not guarantee to obtain optimal results due to simplicity of the experimental run. This argument was supported by Woodall et al., (2003) where they attempted to optimize the features in the medical diagnosis of liver disease using OA and benchmarked the results based on full factorial combinations test. Moreover, Abraham & Variyath (2003) supported this argument in which they attempted to optimize the features for the same case study using Forward Selection procedure. The result showed that via this method, more optimal combination was found compared to optimal combination obtained via OA. However, Forward Selection method has the tendency to suffer from 'nesting-effect' that is, once the variable of interest is chosen, there is no way for it to be omitted (Theodoridis & Koutroumbas, 2009).

In the context of MTS, feature selection is conducted to reduce the number of the original features without having the loss of recognition quality. To meet the objective, orthogonal array (OA) matrix is utilised to search for the optimum set of features while signal-to-noise ratio (SNR) is used to measure the accuracy of prediction via the optimal feature set obtained. This feature selection process in MTS is important because not only it increases recognition accuracy, it also provides computational cost benefits since the resulting recognition will be faster and consume less memory (Thangavel & Pethalakshmi, 2009) due to lesser number of features then the original set.

Taguchi et al. (S. Taguchi, Jugulum, Taguchi, & Wilkins, 2003) recommended a better search algorithm should be incorporated into the MTS methodology to improve MTS performance, thus several attempts to replace the OA with other techniques particularly the Swarm Intelligence-based algorithms such as Particle Swarm Optimization (PSO) (Pal & Maiti, 2010, Reséndiz & Rull-Flores, 2013, Reyes-Carlos, Mota-Gutiérrez, & Reséndiz-Flores, 2018), Ant Colony Optimization (ACO) (Reséndiz, Moncayo-Martínez, & Solís, 2013a) and Bees Algorithm (BA) (Ramlie et al., 2016) have been reported. The reports on the attempts are discussed on the following sub-section.

1.1 Recent Swarm Intelligence-based feature selection studies replacing OA in the Mahalanobis-Taguchi System

The emergence of optimization and feature selection studies adopting Swarm Intelligence (SI) has gained great interest in the literature due to the advantages of collective solution strategies offered by this technique. SI is a collection of metaheuristic search strategy that mimics the collective social behaviour of animals or insects. Particle Swarm optimization (PSO), Firefly algorithm (FA), Cuckoo Search algorithm (CSA), Artificial Fish School algorithms, Bees-inspired algorithms, Bats algorithms to name a few are among the state-of-the-art heuristic search algorithms that fall under the umbrella of SI (Krause, Cordeiro, Parpinelli, & Lopes, 2013). Recent studies have shown that SI search strategies have gained huge appreciations from literature for solving complex and discrete optimization problems with greater success (Blum & Merkle, 2008, Yang, 2010) due to the advantageous of exploration and exploitation search characteristics offered by this technique.

Pal and Maiti (2010) proposed binary PSO algorithm to facilitate MTS optimization scheme replacing the OA in MTS. In spite of using SNR as the objective function, they formulated a mathematical model decision criterion based on minimization of total weighted misclassification (TWM). Misclassification happens when a recognition system wrongly predicts samples out of their true condition (Type I and Type II errors). Hence, to incorporate misclassification in the case of decision making criteria, the associate misclassification probability that minimizes Type I and Type II errors must be considered (Woodall et al., 2003). The proposed method was tested on characterizing the optimum chemical elements of grey cast iron from an Indian foundry shop and the results were compared against the optimum chemical elements obtained via OA based on the final SNR values. The outcome from the study showed that the predictive performance of the proposed method was better compared to OA since it produced higher SNR value. However, validation in terms of performance accuracy based on several testing samples was not being reported. Furthermore, the use of TWM based on misclassification probabilities in MTS seems inappropriate. MTS does not consider any statistical inference on all the sample population prior MTS deployment including the abnormal data since abnormal data are not treated as a separate population (G. Taguchi & Jugulum, 2002, Ghasemi et al., 2015). Nevertheless, to the author's knowledge, replacing OA using PSO algorithm was the first attempt found in the literature to deploy an SI technique in MTS methodology.

Similar optimization approach deploying SI strategy to expedite MTS optimization procedure were demonstrated by Reséndiz et al. (2013b) and Reséndiz and Rull-Flores (2013) to replace OA with Binary Ant Colony Optimization (BACO) and Gompertz Binary Particle Swarm Optimization (GBPSO) algorithms respectively. However, the works aimed at only comparing the respective converging performances against BPSO proposed by (Reséndiz et al., 2013b) in MTS framework based on the number of computational iterations towards the optimum solutions. The studies made no discussions on comparing the performances against OA.

Ramlie et. al (2016) has adopted Bees Algorithm (BA) to replace the OA for feature selection of the MTS. In the study, MD was used as the result assessment metric while the larger-the-better type of SNR was deployed as the algorithm objective function. Character recognition based on Taguchi concepts (exploiting variation and abundance items) was used as the case study on which the comparison between BA and OA performances was made. The results showed a promising discriminant power of the optimized system via BA as compared to OA.

Table 1 summarizes the works found in the literature to replace the OA as the search and optimization strategy in enhancing the MTS methodology. Therefore, in this study, another recent SI optimization technique called *modified*-Bees Algorithm (*m*BA), a modified version from the original BA is introduced into MTS optimization procedure replacing the OA for comparison. The SNR measurement metrics will be retained from the original MTS framework as the objective function for selecting the important features of the system.

The paper is presented as follows, a theoretical overview on fundamental concept of MD and MTS are explained in Section 2. A brief concept of BA as well as the *m*BA and the fusion strategies between the MTS and the newly proposed *m*BA are discussed in Section 3. Section 4 explains the required performance criteria in evaluating the results while Section 5 highlights the case study based on the original Taguchi's liver disease data. The discussion on the results is presented in Section 6 while Section 7 concludes the key findings and contributions of this paper.

Table 1: Studies of MTS by replacing the OA with other SI techniques.

Optimization techniques	**Objective function / feature selection criterion**	**Performance evaluation in comparison to OA/ Conventional MTS**		**Remarks**
		Classification Accuracy	Optimization Speed	
Binary Ant Colony Optimization (BACO) (Reséndiz et al., 2013b)	Minimization of total weighted misclassification	X	X	Speed performances based on number of iterations by BACO and BPSO, no comparison to OA on accuracy and speed was made.
Binary Particle Swarm Optimization (BPSO) (Pal & Maiti, 2010)	Minimization of total weighted misclassification	√	X	Speed performances based on number of iterations by BPSO, no comparison to OA on speed was made.
Gompertz Binary Particle Swarm Optimization (GBPSO) (Reséndiz & Rull-Flores, 2013)	Minimization of total weighted misclassification	X	X	Speed performances based on number of iterations by GBPSO, BACO and BPSO, no comparison to OA was made on accuracy and speed.
Bees Algorithm (Ramlie et al., 2016)	Maximizing signal-to-noise ratio	√	√	Performance measures based on both accuracy and speed by comparisons between BA and OA

Legend: '√' reported in literature; 'X' not reported in literature; '*' proposed method in this work

2 THE CONCEPT OF MAHALANOBIS DISTANCE (MD)

MD is a dimensionless distance measure based on correlation between features and pattern differences that can be analysed with respect to a reference population (Hwang & Park, 2011), as shown in Figure 1. This reference population is called as the normal space. The distance measure termed as the Mahalanobis Scale (MS) and aids the discriminant analysis approach by assessing the level of abnormality of datasets against the normal space.

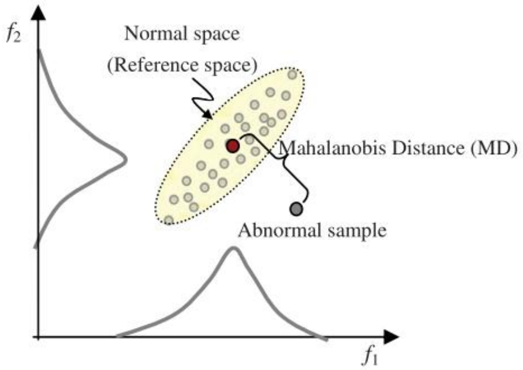


Figure 1: Mahalanobis Distance based on two features (Hwang & Park, 2011)

MD has an elliptical shape (see Figure 1) due to the correlation effect between the features. If there is no correlation, the MD is the same as the Euclidean Distance (ED) that has a circular shape. MD is different from Euclidean Distance since the later does not consider the correlation among the features of the data points.

2.1 Mahalanobis Distance (MD) Formulation

MD is defined as in Equation 1.

$$\boldsymbol{MD_j = D_j^2 = Z_{ij}^T C^{-1} Z_{ij}} \text{ with } \boldsymbol{Z_{ij} = \frac{x_{ij} - m_i}{s_i}} \quad \textbf{(1)}$$

Where;

i = the number of features ($i = 1, 2, \ldots, k$);

j = the number of samples ($j = 1, 2, .., n$);

Z_{ij} = the standardized vector of normalized characteristics of x_{ij};

x_{ij} = the value of the i^{th} characteristic in the j^{th} observation;

m_i = the mean of the i^{th} characteristic;

s_i = the standard deviation of the i^{th} characteristic;

T = the transpose of the vector;

C^{-1} = the inverse of the correlation coefficient matrix.

MD has been well deployed in a broad array of applications (Feng, Hiroyuki, Hidennori, Yuichi, & Hu, 2011, Guo, Yin, Li, & Zhao, 2013) mainly because it is very effective in tracking intervariable correlations in data.

2.2 Mahalanobis-Taguchi System (MTS) procedures

Taguchi extended the MD methodology with his robust engineering concepts to become an efficient and effective strategy for prediction and forecasting in multidimensional systems. In the MTS methodology, the formulation of MD is 'scaled' where the existing MD formulation stated in Equation 1 is divided by a term '$\boldsymbol{k}$' that denotes the number of variables or features of a recognition system. Therefore, the equation for calculating the scaled MD in the MTS methodology becomes:

$$\boldsymbol{MD_j = D_j^2 = \frac{1}{k} Z_{ij}^T C^{-1} Z_{ij}} \quad \textbf{(2)}$$

From this point onwards, the MD computation will be based on Equation (2). The MD offers a statistical measure to diagnose unknown sample conditions with known samples and provides information to make future prediction.

The fundamental steps in the MTS methodology are explained in the next section.

2.2.1 STAGE 1: Construction of measurement scale

To construct a measurement scale, a homogeneous data set from normal observations needs to be collected to build a reference group called the normal group (W. Z. A. W. Muhamad et al., 2017). It is used as a base or reference point in the scale. The collected normal datasets need to be standardized to obtain a dimensionless unit vector followed by the MD computation. Practically, the MD for unknown data is interpreted as the nearness to the mean of the normal group. As a countercheck, the average value of the MDs for the normal group must always be close to unity; therefore they are called the normal space or Mahalanobis Space (MS) (Teshima, Hasegawa, & Tatebayashi, 2012).

The steps for the construction of the MS are outlined below:

a) Calculate the mean characteristic in the normal data set as:

$$\boldsymbol{\overline{x}_i = \frac{\sum_{j=1}^{n} X_{ij}}{n}} \quad \textbf{(3)}$$

b) Then, calculate the standard deviation for each characteristic:

$$\boldsymbol{s_i = \sqrt{\frac{\sum_{j=1}^{n} (X_{ij} - \overline{x}_i)^2}{n-1}}} \quad \textbf{(4)}$$

c) Next, standardise each characteristic to form the normalized data matrix (Z_{ij}) and its transpose (Z_{ij}^T):

$$\boldsymbol{Z_{ij} = \frac{(X_{ij} - \overline{x}_i)}{s_i}} \quad \textbf{(5)}$$

d) Then, verify that the mean of the normalized data is zero:

$$\boldsymbol{\overline{z}_i = \frac{\sum_{j=1}^{n} Z_{ij}}{n} = 0} \quad \textbf{(6)}$$

e) Verify that the standard deviation of the normalized data is one:

$$\boldsymbol{s_z = \sqrt{\frac{\sum_{j=1}^{n} (Z_{ij} - \overline{z}_i)^2}{n-1}} = 1} \quad \textbf{(7)}$$

f) Form the correlation coefficient matrix (C) of the normalized data. The element matrix (c_{ij}) is calculated as follows:

$$\boldsymbol{c_{ij} = \frac{\sum_{m=1}^{n} (Z_m Z_{jm})}{n-1}} \quad \textbf{(8)}$$

g) Compute inverse correlation coefficient matrix (C^{-1})

where:

$$\boldsymbol{C = \frac{Cov(X,Y)}{V(X)V(Y)}} \quad \textbf{(9)}$$

where:

$$\boldsymbol{Cov\,(X,Y) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})} \quad \textbf{(10)}$$

n is the number of samples,

X and Y are two different features being correlated, *X bar* and *Y bar* are the averages among the data in each variable, and *V(X)* and *V(Y)* are the variances of *X* and *Y*.

h) Finally, calculate the MD_j using Equation (2).

2.2.2 STAGE 2: Assessment of the measurement scale

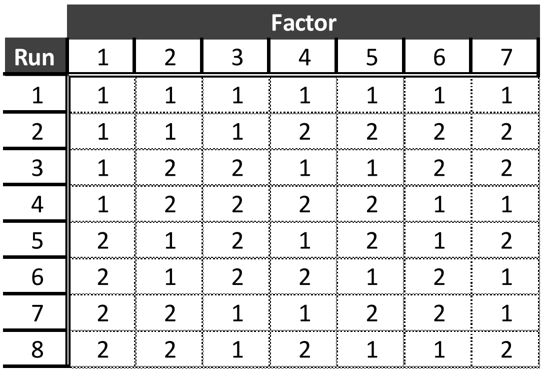
To evaluate the measurement scale, observations outside the MS or abnormal datasets are used. The same mathematical calculation is repeated to calculate the same goal (MD value) using the abnormal sample data. However, the abnormal data is normalized based on the mean, standard deviation and correlation matrix of the normal group. The normal MDs and abnormal MDs are then compared. An acceptable measurement scale should demonstrate significant discrimination between the normal and abnormal MD values.

2.2.3 STAGE 3: Identify significant features

In the third stage, the system is optimized by means of selecting only the features that are known to be significant or 'useful' for the system. This is where the OA and SNR are utilized. The features are assigned to an orthogonal array experimental run of two-level, in which 'used' is signified as level 1 and 'not used' as level 2. The MD for each experiment runs for all 'used' features from each abnormal sample is calculated. The calculated MD values are recorded according to the experimental run. The SNR based on the MD values for all samples is then computed.

2.2.3.1 The role of the orthogonal array (OA) in MTS

Orthogonal array (OA) is a type of fractional factorial design of experiment introduced by C.R. Rao in 1947 (Hedayat, Sloane, & Stufken, 1999). It is different from the traditional fractional factorial DOE in the sense that it tries to balance the combination or interaction of factors equally with the minimum number of experimental runs. In MTS, the orthogonal array structure is represented by Latin symbology as L_a (b^c) where *L* is the Latin Square, *a* is the number of runs, *b* is the number of factor levels and *c* is the number of main factors. Figure 2 illustrates an example of an OA structure for 7 factors with eight runs and two factor levels.

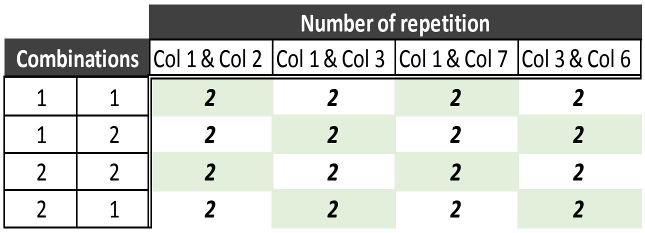


	Factor						
Run	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Figure 2: An example of an OA structure of type $L_8(2^7)$ array.

The name 'orthogonal' is suggested not because of the perpendicular attribute of the structure but rather it is defined as any pairs of columns with the same repetition number of combination of factors (Hedayat et al., 1999). To illustrate further, using the OA in Figure 2 as an example, take a pair between column 1 and column 2, the repetition number of each level of combinations in this column pair is the same (which is twice in this case). The same number of repetitions should be obtained for the rest of the column pairs thus the L_8 (2^7) array depicted by Figure 2 can be said to be orthogonal. Table 2 illustrates the number of repetitions in level combination for another three more column pairs.

Table 2 : The number of repetitions of level combinations.



		Number of repetition			
Combinations		Col 1 & Col 2	Col 1 & Col 3	Col 1 & Col 7	Col 3 & Col 6
1	1	2	2	2	2
1	2	2	2	2	2
2	2	2	2	2	2
2	1	2	2	2	2

The same repetition number of levels (twice) of all two column pairs of this OA structure is obtained; therefore the L_8 (2^7) array, as depicted by Figure 2, can be said to be orthogonal.

In MTS, OAs are used to select the features of importance by minimizing the different combinations of the original set of features. The features are assigned to the different columns of array. Since the features have only two levels, a two-level array is used in MTS as illustrated in Figure 2. For each run of an OA, MDs corresponding to the known abnormal conditions are computed. The importance of features is judged based on their ability to measure the degree of abnormality on the measurement scale (G. Taguchi & Jugulum, 2002). This is where the signal to noise ratio metric is deployed. Further discussion on OA concepts can be found from (G. Taguchi & Jugulum, 2002, Hedayat et al., 1999, Park, 1996 and Phadke, 1989).

2.2.3.2 The role of the SNR in MTS

The signal to noise ratio (SNR) concept which can be considered as the core essence of Taguchi philosophy, is developed by Taguchi who was inspired when he was practicing the engineering profession in a Japanese telecommunication company in the 1950s. In telecommunication context, the SNR captures the magnitude of true information (i.e. signals) after making some adjustment for uncontrollable variation (i.e. noise) (G. Taguchi & Jugulum, 2002). In Taguchi's robust engineering concept, the SNR is defined as the measure of the functionality of the system, which exploits the interaction between control factors and noise factors. A gain in the SNR value denotes a reduction in the variability, hence a reduction in the cost associated with the overall significant factors of interest. Park (1996) and Phadke (1989) provide a detailed description of SNR concepts and its origin of formulation.

In the context of MTS, the SNR is defined as the measure of accuracy of the measurement scale for predicting abnormal conditions (G. Taguchi & Jugulum, 2002). In MTS, a higher value of SNR, expressed in decibels (dB), means a lower prediction error. SNR is used as a metric to assess how significant each variable in the system contributes to the ability to discriminate between the normal and abnormal observations. It could also be used to assess the overall performance of a given MTS system and the degree of improvement after optimization.

The three most commonly used types of SNR in MTS are larger-the-better (LTB), nominal-the-best (NTB), and dynamic (G. Taguchi & Jugulum, 2002, Teshima et al., 2012, G. Taguchi, Rajesh, & Taguchi, 2004). In this study, the larger-the-better SNR will be deployed.

2.2.3.2.1 Larger-the-better SNR

LTB is formulated as in Equation (11) below, where t are the abnormal conditions and D_1^2, D_2^2, ...D_t^2 are the MDs corresponding to the abnormal situations. The SNR (for the larger-the-better criterion) corresponding to q^{th} run of OA is given as:

$$\boldsymbol{SNR} = \eta_q = -\boldsymbol{10log_{10}}\left[\frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{D_i^2}\right)\right] \qquad \textbf{(11)}$$

For each variable X_i, SNR^1 represents the average SNR of level 1 for X_i while SNR^2 represents the average SNR of level 2 for X_i throughout the vertical columns of the OA. Thus, positive gains from Equation (12) constitute useful features while negative gains constitute otherwise. Table 3 illustrates the assessment made using the SNR to evaluate significant factors of the L$_8$ OA structure.

$$Gain = SNR^1 - SNR^2 \qquad (12)$$

Table 3 : An example of useful feature selection using OA (L$_8$ [2^7]) and SNR

	Factor											
Run	1	2	3	4	5	6	7	MD Computation				SNR
1	1	1	1	1	1	1	1	MD_1	MD_2	MD_3	MD_4	SNR_1
2	1	1	1	2	2	2	2	MD_1	MD_2	MD_3	MD_4	SNR_2
3	1	2	2	1	1	2	2	MD_1	MD_2	MD_3	MD_4	SNR_3
4	1	2	2	2	2	1	1	MD_1	MD_2	MD_3	MD_4	SNR_4
5	2	1	2	1	2	1	2	MD_1	MD_2	MD_3	MD_4	SNR_5
6	2	1	2	2	1	2	1	MD_1	MD_2	MD_3	MD_4	SNR_6
7	2	2	1	1	2	2	1	MD_1	MD_2	MD_3	MD_4	SNR_7
8	2	2	1	2	1	1	2	MD_1	MD_2	MD_3	MD_4	SNR_8
Averaging	SNR_{L1}	SNR_{L1}	SNR_{L1}	SNR_{L1}	SNR_{L1}	SNR_{L1}	SNR_{L1}					
	SNR_{L2}	SNR_{L2}	SNR_{L2}	SNR_{L2}	SNR_{L2}	SNR_{L2}	SNR_{L2}					
Subtraction	Gain(+/-)	Gain(+/-)	Gain(+/-)	Gain(+/-)	Gain(+/-)	Gain(+/-)	Gain(+/-)					

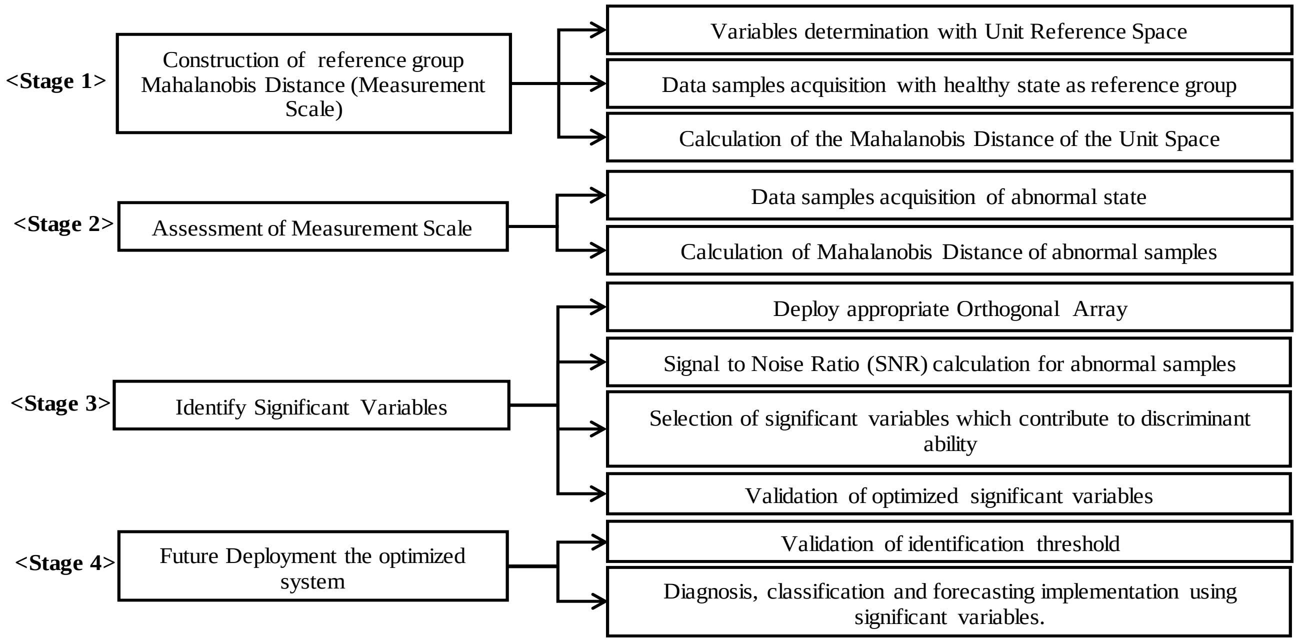

<Stage 1>
Construction of reference group Mahalanobis Distance (Measurement Scale)
Variables determination with Unit Reference Space
Data samples acquisition with healthy state as reference group
Calculation of the Mahalanobis Distance of the Unit Space
<Stage 2>
Assessment of Measurement Scale
Data samples acquisition of abnormal state
Calculation of Mahalanobis Distance of abnormal samples
<Stage 3>
Identify Significant Variables
Deploy appropriate Orthogonal Array
Signal to Noise Ratio (SNR) calculation for abnormal samples
Selection of significant variables which contribute to discriminant ability
Validation of optimized significant variables
<Stage 4>
Future Deployment the optimized system
Validation of identification threshold
Diagnosis, classification and forecasting implementation using significant variables.

Figure 3: Fundamental four stages in MTS methodology

2.2.4 STAGE 4: Future deployment with significant features

The optimized system is then re-evaluated with the abnormal samples to validate the effectiveness of assessing the discriminant power. Once confirmed, the optimized system is used for future application in diagnosis, classification, or forecasting purposes. Figure 3 illustrates the summary of the fundamental stages in MTS.

3 METHODOLOGY

This section presents an overview of Bees Algorithm (BA) methodology developed by Pham et al. (2006) followed by the fusion strategies between MTS and *m*BA.

3.1 Bees Algorithm

Bees Algorithm (BA) was proposed by Pham and his colleagues in 2005 (Pham & Castellani, 2009). BA is an SI technique that provides an optimization solution based on the collective interaction between bee agents within the colony. This collective intelligent mechanism serves as the backbone of the technique at which faster convergence (or divergence) towards the most promising solution is made possible.

BA gets inspiration from the food foraging behaviour of bees to search for the most promising solution to a given optimization problem. Each potential solution in the search space is treated as a food source for the bees. Therefore, each bee carries one possible solution for the problem.

In BA, a population of 'scout bees' randomly search the solutions (food) in the sample space and evaluates the quality of each solution (food) based on a predefined fitness function of the optimization problem. All found solutions by the 'scout bees' are then ranked in either ascending or descending order (depending on the fitness function objective). In a real bee colony, the food quality is evaluated via a 'waggle dance' in which information about the food discovered (the direction of the food source, its distance from the hive and quality of the food) is choreographed (Pham et al., 2006).

The highest ranked solution is chosen as a potential optimum solution and more bees will be recruited to exploit further solution (if any) around its neighbourhood area. The neighbourhood of a solution is called a 'flower patch' in the natural bee colony term. In BA, the neighbourhood is the search landscape area near the best solution found during recruitment. Other lower ranked solutions will also be exploited selectively under a similar strategy.

Despite the exploitation being performed in the neighbourhood area, the remaining scout bees will be assigned for global random searches to explore and locate (if any) other promising solutions better than the one that has been exploited before. This explorative search strategy is crucial to avoid the scout bees' population being trapped at a local optimum during the search. Finally, a new set of 'scout bee' population is formed comprising the bees with the best solutions for all exploited sites discovered so far. A similar process (random search, neighbourhood exploitation, and global exploration) is repeated until a global best with a promising solution is found or the search limiting criterion is met. Pham *et al.* characterized the behaviour of the bees in foraging for food into an algorithm called the Bees Algorithm.

BA requires several parameters prior to deployment which is portrayed in Table 4. The number of scout bees needed for the global search process is labelled as '*ns'*. Out of the preliminary solutions (food sites) found by the scout bees, the number of elite sites (elite solution) labelled as '*ne'* and the number of the best site (best solution) labelled as *'nb'* is formed. Each elite and best solutions obtained previously will be assigned a number of recruited bees for neighbourhood search process (exploitation purpose) labelled as '*nre'* for the elite sites and *'nrb'* bees for the remaining best sites respectively. The size or boundary of the neighbourhood search for the exploitation process by both recruited bees mentioned above is labelled as '*ngh*' and the final parameter that is required control the iteration limit of the algorithms is labelled as '*stlim*'.

Table 4: Basic Bees Algorithm Parameters

ns	number of scout bees
ne	number of elite sites
nb	number of best sites
nre	recruited bees for elite sites
nrb	recruited bees for remaining best sites
ngh	intinial size of the neigborhood
stlim	limit or stopping criterion

Figure 4 illustrates the basic flow chart of BA proposed by Pham *et al.* (2006) where it incorporates the exploration and exploitation behaviour of the bees in nature into the algorithm which is described in the following sub-sections.

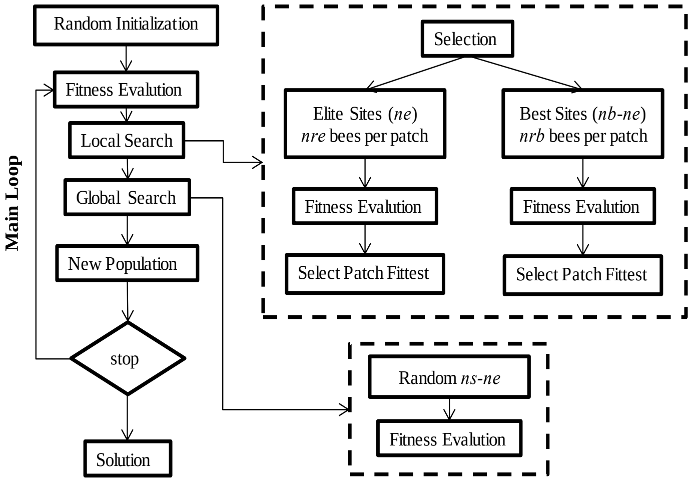


Figure 4 : Flowchart of the basic Bees Algorithm

BA has been widely used and solved various kind of optimization problems including combinatorial and discrete optimization problems, functional and continuous optimization problems, optimization in the design of mechanical components, multi-objective optimization problems as well as

NP-Hard problems such as the Generalized Assignment Problem (GAP) (Özbakir, Baykasoğ lu, & Tapkan, 2010; Tsai, 2014).

3.2 The Proposed *modified*-BA (*m*BA) and its fusion strategy in the MTS

3.2.1 Problem Understanding

In the context of optimization literature, optimization problems can be divided into two type namely continuous and discrete (combinatorial or binary) optimization problems. Unlike the continuous optimization problem, a discrete optimization problem requires an update of its vector solution that represents the combinatorial problem by preserving the validity of the combination of the solution (Krause et al., 2013). The latter can be further grouped into four main classes of problems namely the ordering problems, the assignment problems, the grouping problems and the subset (feature selection) problems (Engelbrecht, 2006), of which the MTS optimization problem falls under the latter category. Optimization problem in the context of MTS, is considered a discrete optimization problem since the optimization solution sought via the MTS comes in a reduction number of combinatorial features from the original feature set. To fuse a swarm intelligence algorithm such as the *m*BA into this discrete nature of optimization problem such as the MTS, an encoding process to a discrete dimension which often called as *discretization* need to be adapted.

3.2.2 Discretization and Problem Formulation

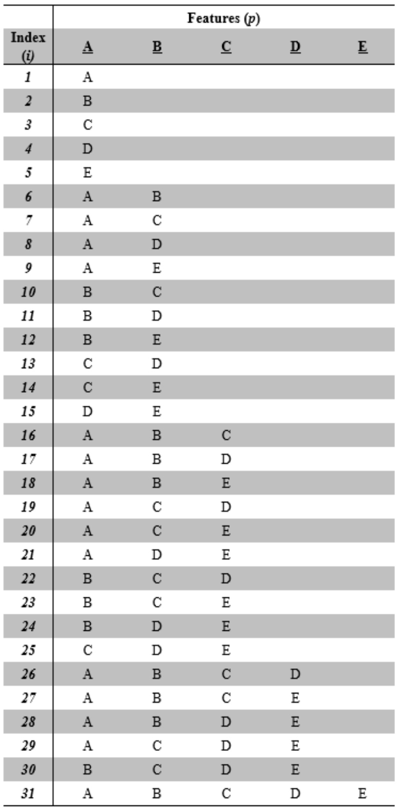
There are several discretization methods that have been reported in the literature. In the context of fusing swarm intelligence algorithm into the MTS discrete optimization problem, (Pal & Maiti, 2010) and (Reséndiz & Rull-Flores, 2013) deployed a discretization method using a sigmoid function to characterized the Particle Swarm Optimization algorithm into a binary characterization of the solution vector. (Reséndiz et al., 2013b) adopted similar discretization method when they characterized Ant Colony Optimization algorithm into the MTS framework. Ramlie et al. (2016) however, adopted a different type of discretization method called the smallest position value (SPV) discretization. Unlike the sigmoid function discretization which encodes the vector solution in binary (0s and 1s) form, SPV maps the position of the solution vectors by placing the lowest index of the solution component (i.e. smallest combination of solutions) as the first item on a permutated or combinatorial landscape followed by the second lowest and so on until the largest combination. This discretization technique creates an integer vector solution by indexing the position of all possible combination of the solution in an enumeration list (Krause et al., 2013).

Hence, the goal of a subset problem (SSP) is similar to *feature selection* goal (in the pattern recognition context) which is to select the best subset of n_s from a set of n (n is the full feature set of a system such that $n_s \subseteq n$) items that a given objective function is optimized (minimize or maximize) under this subset of n_s without violating the problem's constraints.

Traditionally, in the context of the feature selection problem of the MTS, orthogonal array (OA) is used to search the optimum subset of features while signal-to-noise ratio (SNR) is used to measure the accuracy of a prediction made by the optimal feature subset obtained. However, since SI algorithms such as the *m*BA is to be deployed replacing the OA in this work, the subset selection of MTS is characterized as follows:

An enumeration group defined as S: 2^p-1 is constructed that lists all possible combinations of the total numbers of the decision variables, ***p***. Each subset combination is labelled as subset X_i where i = 1 to 2^p-1. Each X_i carries a unique subset of combination such that subset $X_i \subseteq S$. This enumeration integer list bounded by i = 1 to 2^p-1 provides the lower and the upper boundary limit in the global search (exploration) landscape.

Figure 5 illustrates further this concept explained above in graphical illustration taking an example of p = 5 number of features. As shown in Figure 5, with p = 5 as the total number of total features of the original system, for instance, a total of 31 number of possible unique combinations (25-1 = 31) is computed. Therefore, the integer index ranges from the first combination until the 31st set of combination, forms the search lower and upper boundaries of the swarm intelligence algorithm (i.e. the BA) within the optimization search landscape. From the index i range, a solution of "X1" consists a subset solution of 'A', a solution of "X17" consists a subset solution of 'A', 'B', 'D' while a solution of "X31" consists a subset solution of 'A', 'B', 'C', 'D' and 'E' such that "X1" =[A], "X17" =[A, B, D] and "X31" =[A, B, C, D, E] to form their respective vector components.



	Features (*p*)				
Index (*i*)	A	B	C	D	E
1	A				
2	B				
3	C				
4	D				
5	E				
6	A	B			
7	A	C			
8	A	D			
9	A	E			
10	B	C			
11	B	D			
12	B	E			
13	C	D			
14	C	E			
15	D	E			
16	A	B	C		
17	A	B	D		
18	A	B	E		
19	A	C	D		
20	A	C	E		
21	A	D	E		
22	B	C	D		
23	B	C	E		
24	B	D	E		
25	C	D	E		
26	A	B	C	D	
27	A	B	C	E	
28	A	B	D	E	
29	A	C	D	E	
30	B	C	D	E	
31	A	B	C	D	E

Figure 5 : Illustration of the discretization approach of this study taking example of *p* = 5 features

Figure 5 illustrates the global search process conducted by the scout bees in the global search landscape bound by integer i (in this case it is between 1 to 31) which assigned to solution component of X "⊆" [A,B,C,D,E]. As shown in Figure 5, take the number of scout bees (ns) equals to 5 for instance, all the assigned scout bees will randomly 'fly' to find the promising 'food locations' or solutions bounded by the search index ranges between X1 to X31 (since p=5 as in this illustration). Once the bees found their candidate of solutions, take the first bee for instance, it 'landed' on X6 of the 'food location' (one of the solutions' candidates) index which consists of solution vector of features 'A' and 'B', the bee started to compute the MD values based on the respective features as well as its corresponding SNR value. The SNR value obtained from this computation represents the solution value of the search found by the particular bee which carries the feature selection of variables 'A' and 'B' as its current optimized features.

Take the last bee for instance, it randomly explores the global search landscape and 'landed' on a 'food location' of X30 (one of the solutions' candidates) index which consists of solution vector of features 'B', 'C', 'D' and 'E', the bee will compute the MD values based on the features that it has currently found followed by computation of its corresponding SNR value. The SNR value obtained from this computation represents the current solution value of the current search found by the bee which carries the feature selection of variables 'B', 'C', 'D' and 'E' as its current optimized features. The other bees (as illustrated in Figure 6) follow a similar global search as well as MDs and SNRs computational processes.

Thus, the objective of the problem which was characterized in this study is to find the integer i on any X_i that carries the optimum subset of the decision variables which maximizes the objective function $f(X_i^{t+1})$ such that:

$$f(X_i^{t+1}) > f(X_i^{t}) \ , \qquad \mathbf{13}$$

For each $X \subseteq S$, i = 1 to 2^p-1 and t is the time step,

Subjected to:

$f(X_i^t) > f(X^0_{original\ set})$;

$i > 0$;

$i \leq 2^p - 1$; where $\boldsymbol{p}$ is the number of total decision variables in the original system.

Therefore, the objective function $f(X_i^t)$ in this problem is formulated based on Equation (11) which is larger-the-better SNR.

3.3 The Proposed BA Neighbourhood Search Structure

In the previous section, we have presented the 'exploration' of the scout bees through an integer discretization of the global search landscape. In this section, the 'exploitation' process which is known as the neighborhood search in a bee colony to intensify the search process in finding better solutions is presented.

The neighborhood (local) search structure is built similar to the global search architecture which was previously mentioned, however, the only difference between the global search and the neighborhood search lies on the selected features that the bees need to be exploited. The number of decision variables or the features is based on the current best subset found so far. Thus, the neighborhood search landscape can be represented as a group of combination sets defined as N: $2^{\boldsymbol{k}}$-1 where $\boldsymbol{k}$ is the number of decision variables found so far such that $k \subseteq p$, which assigns to the group set of N∈S. Each subset combination in the neighborhood search area is labeled as subset X_j with j = 1 to $2^{\boldsymbol{k}}$-1 such that subset $X_j \subseteq N \subseteq S$. Hence, a locally maximum solution $f(X_j)$ (or minimum solution depending on the objective of the problem) could be determined from this neighborhood structure. Since this study is intended to maximize the objective function, a locally maximum solution with respect to a neighborhood structure N is a solution such that $\forall X_j^t \in N : f(X_j^t) \geq f(X_i^t)$ (Blum & Roli, 2003). Figure 7 provides a graphical illustration of an example of the neighborhood search process (the one on the right side of the figure) of the BA in this study.

.

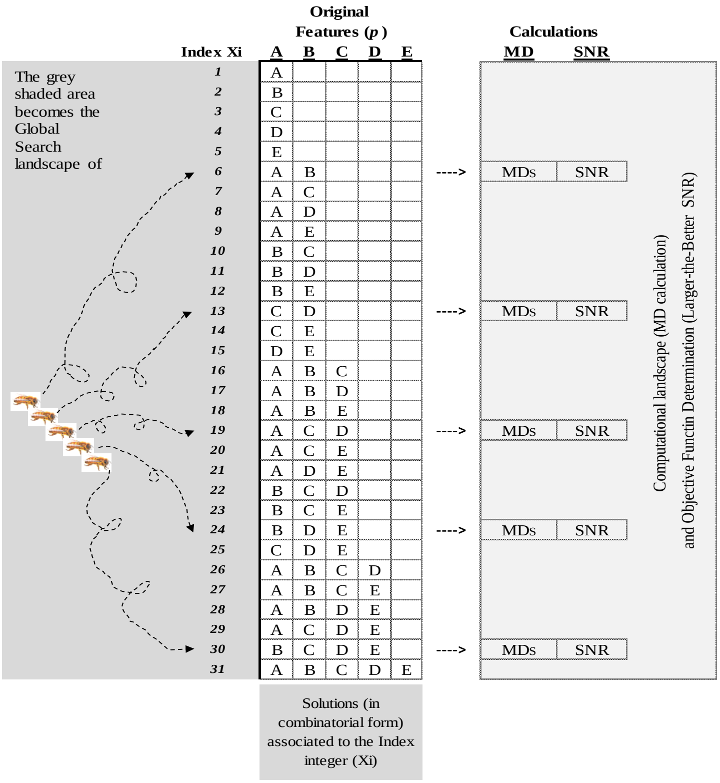

Original
Features (p)
Index Xi
A B C D E
Calculations
MD SNR
The grey shaded area becomes the Global Search landscape of
MDs SNR
Computational landscape (MD calculation) and Objective Functin Determination (Larger-the-Better SNR)
Solutions (in combinatorial form) associated to the Index integer (Xi)

Figure 6 : The proposed global search architecture of the study

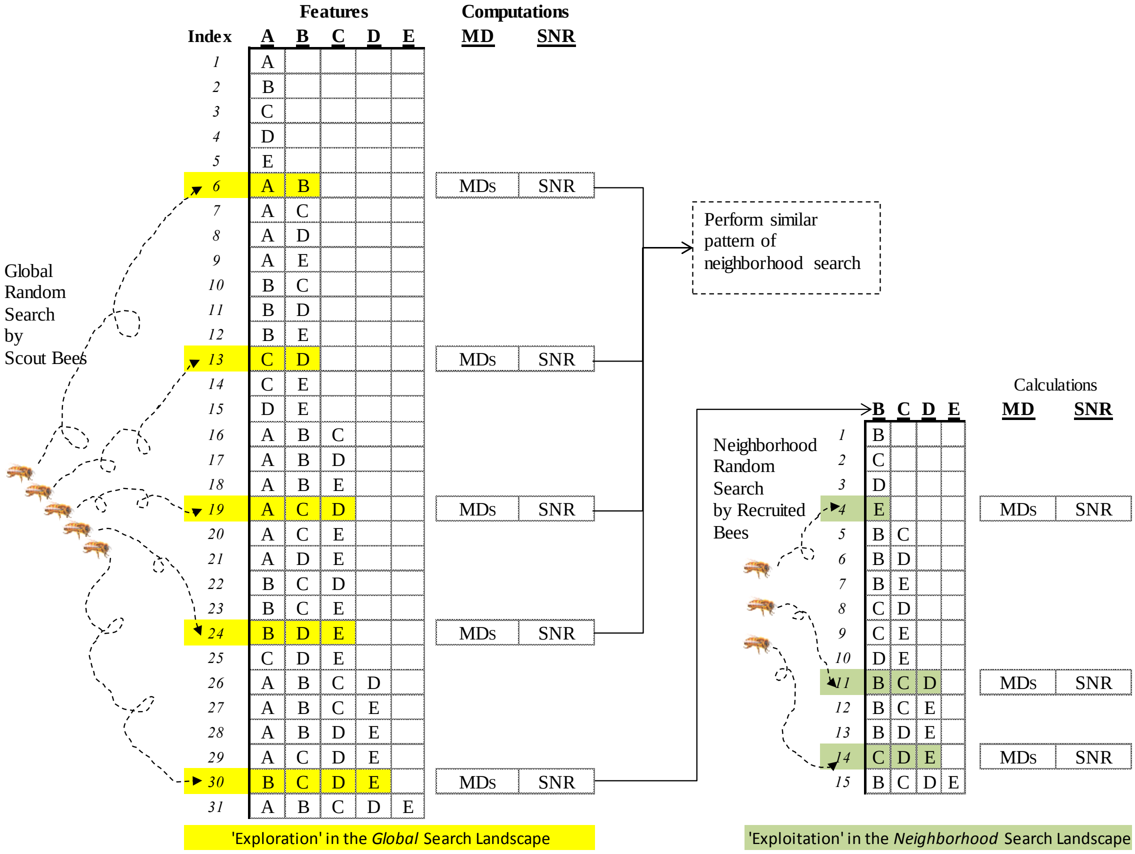

Features
Computations
Index
A B C D E
MD SNR
MDs SNR
Global Random Search by Scout Bees
Perform similar pattern of neighborhood search
Calculations
B C D E
Neighborhood Random Search by Recruited Bees
'Exploration' in the Global Search Landscape
'Exploitation' in the Neighborhood Search Landscape

Figure 7 : Neighbourhood search architecture proposed in this study

The graphical representation of Figure 7 is the extension of the global search process illustrated by Figure7 previously in which the neighborhood search example is initiated from the current global solution found by the last bee (the one that landed on X_{30} solution index). In this example (see Figure), the solution found by this particular scout bee represented by the solution index of X_{30} that consists of 'B', 'C', 'D' and 'E" as its solution candidates will be further discretized under similar integer discretization process of the global search architecture presented in the previous section to form a neighborhood search landscape for the exploitation purposes. As depicted in Figure, with four features ('B', 'C', 'D' and 'E') for example, a new combination list consisting of 15 different combination sets of 'B', 'C', 'D' and 'E' is formed. The newly formed neighborhood search landscape comprises another solution index ranging from i=1 to i=15 based on the 2^4-1=15 computation.

If the global search landscape is randomly explored by the scout bees (*ns*), the neighborhood search landscape such as this example, however, will be randomly exploited by the recruited bees (*nrb*). Based on the example illustrated in Figure 7, 3 recruited bees were assigned to exploit potential solutions in the neighborhood search landscape by which similar MDs and SNRs computational procedures as conducted by the scout bees in the global search landscape were performed. Take the second recruited bees for instance (see the right-hand side of Figure7), the recruited bee was 'landed' on a 'food location' denoted as X_{11} which comprises a solution candidate of 'B', 'C' and 'E'. The particular recruited bee then continued to compute the MD value based on this three feature sets followed by its corresponding SNR computation. The other recruited bees (as illustrated in Figure) follow a similar neighborhood search processes as well as MDs and SNRs computational procedures.

In this proposed bee's algorithmic process, if the solution found by any of the recruited bees in the neighborhood search landscape is better than the solution found by the scout bees in the global search landscape, the solution found by the bees during the neighborhood search will be adopted as the current best solution found so far. In other words, should the recruited bees fail to find a better solution than that of the scout bees, the solution obtained by the scout bees in the global search landscape will be retained as the current best solution found so far.

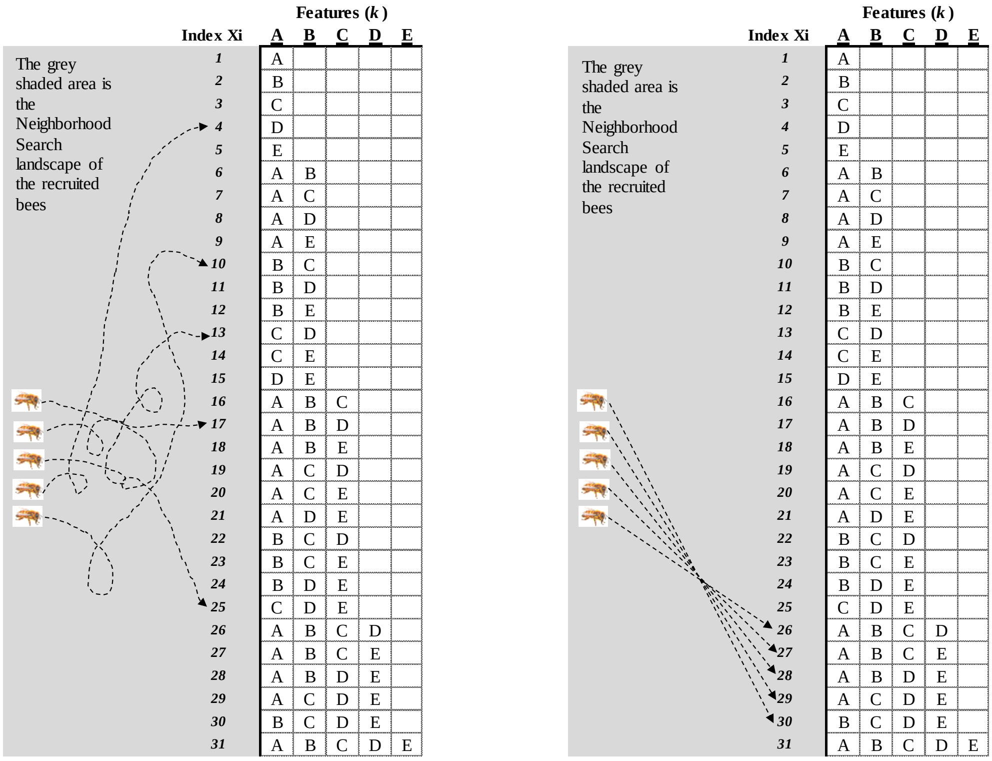
3.3.1 Backward Selection Search vs Random Search

Despite a random means for the neighborhood search by the recruited bees, a *backward selection* neighborhood search technique is also proposed in this study. The random search by the recruited bees in the Bees Algorithm proposed by Pham et.al (2005) is adapted to the natural behavior of the bee colony in nature while exploiting potential high quality of nearby food sources that surrounds the promising flower patch recommended by the scout bees. In the context of Bees Algorithm, random search has the advantage of finding the promising solution more quickly but only by chance, and it poses a drawback in which random search has no control over the recruited bees to not 'landing' on the same 'food location' which had been visited by the previously recruited bees. Thus, they may share the same solution over the same search period without any unique solution from one another. This may increase the search effort and reduce the possibility to exploit other promising solutions in the neighborhood search landscape.

Some may question as to why not utilize a *forward selection* technique instead? In the context of feature selection problem, reports from literature have suggested that from computational point of view, backward selection technique is found to be more computationally efficient than that of the forward selection one in finding the optimal (or near-optimal) solution (Jain, Duin, & Mao, 2000; Kumar & Minz, 2014; Theodoridis & Koutroumbas, 2009). In the proposed backward selection search technique for the neighborhood search process of the BA, the recruited bees will be firstly directed to the solutions with a higher order of combinations of features, up to the least order of combinations. Such search dedication can facilitate the recruited bees to visit a unique food location from one another hence increase the possibility to explore other unexploited sites which eventually enhance the search effort in finding a more promising solution faster.

To understand the difference between the random search and the proposed backward selection search, Figure7 provides an example in graphical illustration of the difference between the two. Figure 8(a) illustrates the random search behavior while Figure 8(b) illustrates the proposed backward selection search behavior of the recruited bees in the neighborhood search process. From Figure8(a), with a *k*=5 number of neighborhood feature for instance, a total of 31 number of possible combinations (possible candidate of solutions) was generated based on the $2^{k=5}$-1 computation which forms a neighborhood search landscape between the index *i=1* to *i=31*. A random search as portrayed in Figure8(a) behaves in such a way that all the recruited bees (*nrb=4* in this case) will 'fly' to the food location anywhere between the solution index of *i=1* to *i=31* in random order. Take the last recruited bee as portrayed in Figure8(a) for instance, it 'landed' on the solution index with lower index number ($X_i=10$) as compared to it previously recruited mate, take the first recruited bee for instance of which it 'landed' on a higher index number (i.e $X_i=17$). Not to mention as previously stated that the last recruited bee (see Figure 8(a)) could also be potentially 'landed' on the same 'site' as the first recruited bee since it searches the 'food location' randomly. A similar condition could also be posed by the other recruited bees under random search behavior.

While in Figure 8(b), unlike random search behavior, each recruited bee under backward selection search strategy is dedicatedly assigned to explore potential promising food location decendingly within the neighborhood search landscape starting from the highest index number (i.e. $X_i=30$ down to $X_i=26$ for this case). The first recruited bee as portrayed in Figure8(b) will exploit the solution index consisting feature solution of 'B', 'C', 'D' and 'E' since it carries the largest index number ($X_i=30$) while the last recruited bees will exploit feature solution 'A', 'B', 'C' and 'D' since it carries the least solution index number ($X_i=26$) among the other solution index in this example. Thus, with such dedicated search processes, exploitation redundancies by the recruited bees could be avoided.


The grey shaded area is the Neighborhood Search landscape of the recruited bees
Features (*k*)

Index Xi	A	B	C	D	E
1	A				
2	B				
3	C				
4	D				
5	E				
6	A	B			
7	A	C			
8	A	D			
9	A	E			
10	B	C			
11	B	D			
12	B	E			
13	C	D			
14	C	E			
15	D	E			
16	A	B	C		
17	A	B	D		
18	A	B	E		
19	A	C	D		
20	A	C	E		
21	A	D	E		
22	B	C	D		
23	B	C	E		
24	B	D	E		
25	C	D	E		
26	A	B	C	D	
27	A	B	C	E	
28	A	B	D	E	
29	A	C	D	E	
30	B	C	D	E	
31	A	B	C	D	E

a

Index Xi	A	B	C	D	E
1	A				
2	B				
3	C				
4	D				
5	E				
6	A	B			
7	A	C			
8	A	D			
9	A	E			
10	B	C			
11	B	D			
12	B	E			
13	C	D			
14	C	E			
15	D	E			
16	A	B	C		
17	A	B	D		
18	A	B	E		
19	A	C	D		
20	A	C	E		
21	A	D	E		
22	B	C	D		
23	B	C	E		
24	B	D	E		
25	C	D	E		
26	A	B	C	D	
27	A	B	C	E	
28	A	B	D	E	
29	A	C	D	E	
30	B	C	D	E	
31	A	B	C	D	E

b

Figure 8 (a) : Random search example; (b) backward selection search example

3.4 The Stopping Criterion

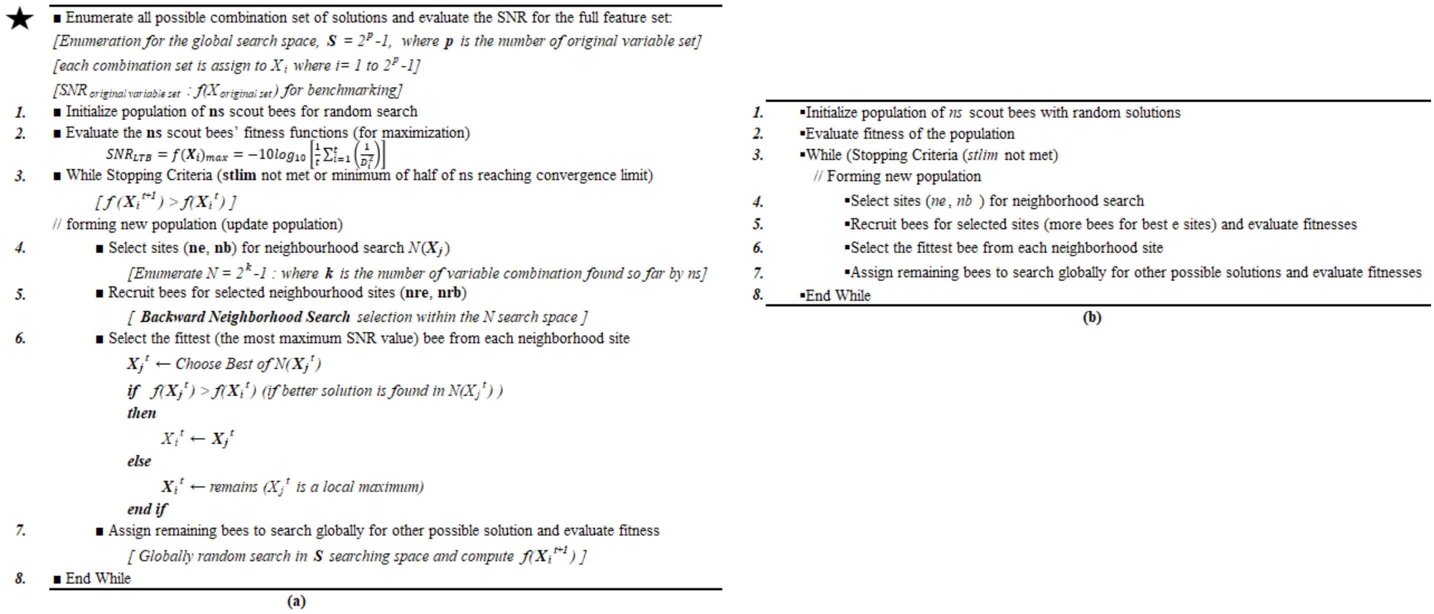
To set the stopping criterion for minimizing objective function is common since it is based on value approaching the limit of zero value. However, setting the limit for maximizing an objective function is slightly challenging as the maximum value of the objective function as in this particular study is unknown (it could be any maximum value). This is where the significance of population-based solutions offered in swarm intelligence technique like BA is beneficial. The stopping criterion will be set when the optimum solution is reached with the same value by (almost) all population of the scout bees in the algorithm. When the saturated values have been reached, the population of the scout bees can no longer find other better solutions than what they have found so far hence denoting it as the final solution. Alternately, if most of the scout bees are not converging to the final solution, the algorithm is stopped when it reaches the iteration limit.

3.5 Pseudo Code of the Proposed Algorithms

Finally, Figure 9(a) illustrates the pseudo code of the proposed algorithm which incorporates all the designed elements mentioned in Section 3.2.1 until Section 3.4 previously. The original pseudo-code of the BA (in its simplified form) proposed by Pham *et al.,* (2006) was also shown on the right-hand side of the figure labeled as Figure 8(b). Note that the modified-Bees Algorithm proposed in this work followed similar algorithmic steps (i.e. 8 lines) with the conventional Bees Algorithm proposed by Pham *et al.,* (2006) to preserve its originality. The exception was given to the top coding line of the proposed pseudo code of this work (see the 'star' marking on the top coding line of Figure 9(a)) in which discretization process mentioned in Section 3.2.2 was included at the initial stage to adapt the algorithm with the discrete nature of optimization problem of MTS. The original pseudo-code as depicted in Figure 9(b) does not have such requirement since it is originally designed for a continuous optimization problem.

From Figure 9(a), in line 1 of the pseudo-code, an initial population of *n* number of scout bees will be assigned to visit potential 'food' sites and to search for an initial solution within the discretized global search landscape randomly. The fitness (the larger-the-better SNR in this case) of the visited sites (represented by the solution index containing feature candidates in this case) found by the scout bees are evaluated in line 2 of the pseudo code. While the stopping criterion mentioned in Section 3.4 is not met, the algorithm continues in line 4 where the bees with the highest fitness (i.e. the highest SNR value in this case) are chosen as 'selected bees' and the respective sites visited by them are chosen for neighborhood search in a group of recruited bees assigned respectively. Thus, several integer discretization of the neighborhood search spaces are formed based on the solution found so far from each scout bee before the neighborhood exploitation could be initiated. Then in line 5 of the proposed pseudo-code, with the newly formed neighborhood search landscapes, each group of the recruited bees started to exploit for a more promising solution in their respective neighborhood search spaces, assigning more recruited bees for the best elite sites. It is only in this line 5 of the proposed pseudo code that the difference between the BA and the mBA is evident. For the BA, the recruited bees exploit the neighborhood sites randomly while for the *mBA*, the recruited bees visit the neighborhood sites using backward selection technique mentioned in sub-section 3.3.1 above.

In line 6, on each respective neighborhood site visited by the respective recruited bees, only one bee which carries the best fitness (i.e. highest SNR value in this case) among them on each neighborhood site will be selected to form the next bee population for the succeeding iteration. Otherwise, the previous solution recommended by the initial scout bees will be maintained. In line 7, the remaining bees in the current bee population are assigned randomly around the global search space (i.e. around the integer discretization of the global search landscape formed at the initial stage of the algorithm) scouting for a new potential solution. Finally, in line 8, the algorithm repeats the similar exploitations (intensifications) and explorations (diversifications) processes starting back from line 3 of the algorithm until the stopping criterion as mentioned in Section 3.4 is finally met.



```
★    ■ Enumerate all possible combination set of solutions and evaluate the SNR for the full feature set:
     [Enumeration for the global search space, $S = 2^p - 1$, where $p$ is the number of original variable set]
     [each combination set is assign to $X_i$ where i= 1 to $2^p - 1$]
     [$SNR_{original\ variable\ set}$ : $f(X_{original\ set})$ for benchmarking]
1.   ■ Initialize population of ns scout bees for random search
2.   ■ Evaluate the ns scout bees' fitness functions (for maximization)
           $SNR_{LTB} = f(X_i)_{max} = -10 log_{10}\left[\frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{D_i^2}\right)\right]$
3.   ■ While Stopping Criteria (stlim not met or minimum of half of ns reaching convergence limit)
           [$f(X_i^{t+1}) > f(X_i^t)$]
     // forming new population (update population)
4.         ■ Select sites (ne, nb) for neighbourhood search $N(X_j)$
               [Enumerate $N = 2^k - 1$ : where $k$ is the number of variable combination found so far by ns]
5.         ■ Recruit bees for selected neighbourhood sites (nre, nrb)
               [ Backward Neighborhood Search selection within the N search space ]
6.         ■ Select the fittest (the most maximum SNR value) bee from each neighborhood site
               $X_j^t \leftarrow$ Choose Best of $N(X_j^t)$
               if $f(X_j^t) > f(X_i^t)$ (if better solution is found in $N(X_j^t)$ )
               then
                   $X_i^t \leftarrow X_j^t$
               else
                   $X_i^t \leftarrow$ remains ($X_j^t$ is a local maximum)
               end if
7.         ■ Assign remaining bees to search globally for other possible solution and evaluate fitness
               [ Globally random search in $S$ searching space and compute $f(X_i^{t+1})$ ]
8.   ■ End While
```

(a)

```
1.   ▪Initialize population of ns scout bees with random solutions
2.   ▪Evaluate fitness of the population
3.   ▪While (Stopping Criteria (stlim not met)
       // Forming new population
4.         ▪Select sites (ne , nb ) for neighborhood search
5.         ▪Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses
6.         ▪Select the fittest bee from each neighborhood site
7.         ▪Assign remaining bees to search globally for other possible solutions and evaluate fitnesses
8.   ▪End While
```

(b)

Figure 9: The pseudo code of the proposed modified-BA (*m*BA).

The modified-BA is deployed replacing the conventional OA as the optimization technique in MTS. The modified element proposed in this paper is elaborated on section 3.2.4. Figure 10 illustrates the conceptual idea behind the strategy.

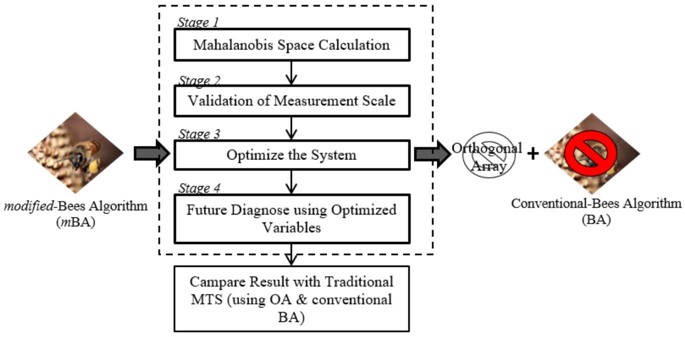

Stage 1
Mahalanobis Space Calculation
Stage 2
Validation of Measurement Scale
Stage 3
Optimize the System
Stage 4
Future Diagnose using Optimized Variables
Campare Result with Traditional MTS (using OA & conventional BA)
modified-Bees Algorithm (*m*BA)
Orthogonal Array
+
Conventional-Bees Algorithm (BA)

Figure 10: Conceptual flow chart of the study

3.5.1 MTS Parameters

In this fusion strategy, the conventional scaled MD is used as the metric to build the baseline multivariate space using Equation (2) while the larger-the-better (LTB) SNR is used as the objective function to be solved as denoted in Equation (11). Hence, the objective of *mBA* algorithm in the fusion strategy is to determine the maximum SNR value among the different sets of features using Equation (11). The set of features with the largest SNR value is the potential solution.

4 PERFORMANCE CRITERIA

In this study, three performance criteria are used to evaluate the performance of the proposed algorithm namely SNR gain as proposed by Taguchi and Jugulum (G. Taguchi & Jugulum, 2002), as well as classification error and computational cost (i.e. running time) as suggested by Jain et al. (Jain et al., 2000).

4.1 SNR Gain

In MTS, the SNR is not only used to identify useful features, but it can also be used to measure improvement in the functionality of the system (G. Taguchi & Jugulum, 2002) based on the gain in the SNR value. An SNR gain is defined as a value having a positive real number when the SNR value of the original system is subtracted by SNR value of the optimized set. A positive gain value denotes an improvement in the optimized system. The higher the positive value, the better the optimized system is in terms of its functionality and performance compared to the original (not optimized) system. A positive SNR gain relates to variability (noise) reduction in the optimized system (G. Taguchi & Jugulum, 2002). Hence, an accurate recognition and prediction decision is obtained with less computing costs. Table 5 illustrates an example of an SNR gain computation.

Table 5: Example of SNR Gain calculation with a positive gain

SNR (Optimal system)	13.98 dB
SNR (Original system)	11.58 dB
Gain	**2.46 dB**

4.2 Classification accuracy and computational cost

The classification error or simply the error rate is another way of measuring the performance of the optimized systems. In practice, the error rate of a recognition system must be estimated from all the available samples which are split into training and test sets (Jain et al., 2000) and are independent from one another. The recognition system is first constructed using the training samples, and then it is evaluated based on its classification performance on the test samples. The percentage of misclassified test samples is taken as an estimate of the error rate. Another performance measure which often overlooked in practice is the cost of measuring features or the computational requirements of the decision rule (Jain et al., 2000). Therefore, in this study, this requirement is also implemented as the performance measure.

4.3 Exhaustive Search

An exhaustive search algorithm was conducted as a comparative study (an analogy to a full factorial in design of experiment context). The exhaustive search technique is guaranteed to find the true optima of any given optimization problem (Jain et al., 2000) since it will search and evaluate each and every single possible solution. However, the drawback of this strategy is that it requires large computational efforts (i.e. long computing time).

5 CASE STUDY

The performance evaluation in terms of the computing speed is performed on the medical diagnosis of liver disease data provided by Taguchi and Jugulum (G. Taguchi & Jugulum, 2002). In the following section, the deployment of the modified-Bees Algorithm in MTS using the data is presented.

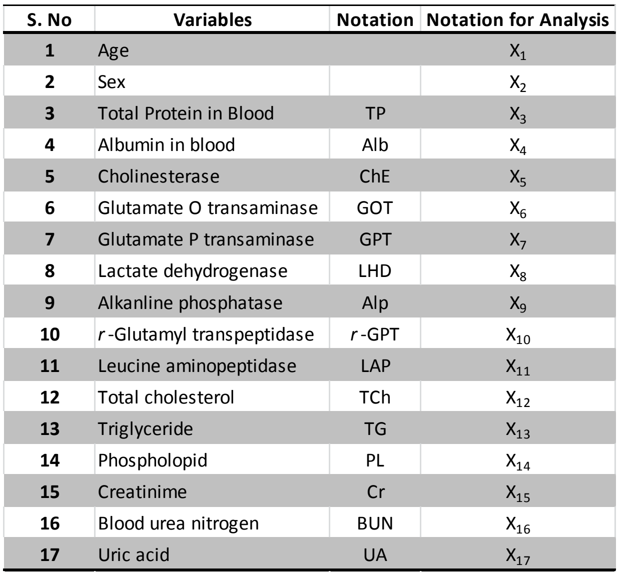
5.1 Medical diagnosis of liver disease data

Liver disease data is a data that was originally collected and used for MTS analysis by Dr. Genichi Taguchi himself during his initial work on MTS. This data can be considered as a renowned data when it comes to evaluating MTS performances since it has been applied by various researchers in evaluating and analysing MTS performances (G. Taguchi, Chowdhury, & Wu, 2005; G. Taguchi & Jugulum, 2002; Teshima et al., 2012).

The story behind the data came over nearly 30 years ago when Dr. Genichi Taguchi working together with Dr. Tatsuji Kanetaka of Tokyo Tenshin Hospital on which they embarked on a joint study of liver disease diagnosis. The result of the study was made public in 1987 and the data was published in various publications as well as being used for several MTS-related study purposes.

The data contains observations of a healthy group as well as the abnormal on 17 features as shown in Table 6.

Table 6: Features in the liver disease diagnosis and notations for the analysis



S. No	Variables	Notation	Notation for Analysis
1	Age		X_1
2	Sex		X_2
3	Total Protein in Blood	TP	X_3
4	Albumin in blood	Alb	X_4
5	Cholinesterase	ChE	X_5
6	Glutamate O transaminase	GOT	X_6
7	Glutamate P transaminase	GPT	X_7
8	Lactate dehydrogenase	LHD	X_8
9	Alkanline phosphatase	Alp	X_9
10	*r*-Glutamyl transpeptidase	*r*-GPT	X_{10}
11	Leucine aminopeptidase	LAP	X_{11}
12	Total cholesterol	TCh	X_{12}
13	Triglyceride	TG	X_{13}
14	Phospholopid	PL	X_{14}
15	Creatinime	Cr	X_{15}
16	Blood urea nitrogen	BUN	X_{16}
17	Uric acid	UA	X_{17}

The healthy group (MS) is constructed based on observations of 200 people (healthy), who do not have any health problems together with 17 abnormal conditions (unhealthy). These data acts as the training data for the construction of initial MT system. Figure illustrates the MD distributions of the training data with a clear separation between the healthy and unhealthy samples (see the enlarged segment). While a total of 60 samples (other than the training samples) are taken as the testing samples (G. Taguchi, Chowdhury, & Wu, 2001).

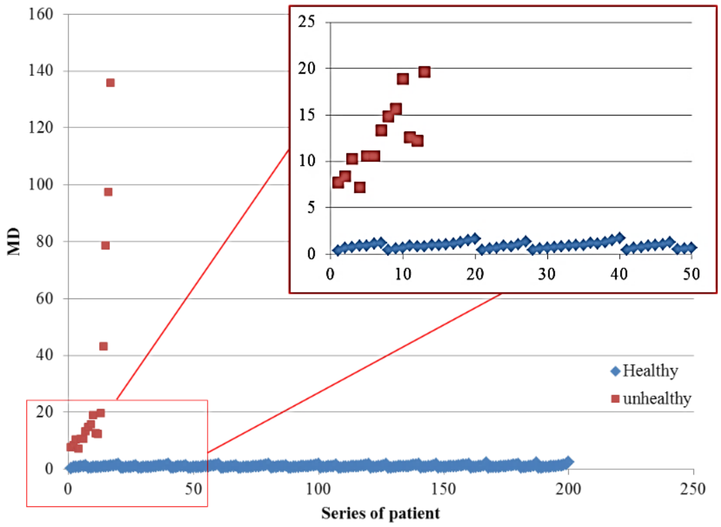


Figure 11: Scatter plot of MD distributions of the training data samples.

Table 6: Summary of the case study

Case study	No. of system features	Threshold value	Size of training data	Size of testing data

Liver disease	17	Healthy ≤ 9
		Unhealthy > 9
			Healthy = 200
			Unhealthy = 17
			Total = 217
				Healthy = 43
				Unhealthy = 17
				Total = 60

Table 6 provides a data summary of the case study in terms of the following: (i) number of system features; (ii) system categories; (iii) size of training dataset; and (iv) size of testing dataset.

5.2 Orthogonal array structure

Since there are 17 features in the case study, an L_{32} (2^{31}) array will be used. Table 7 illustrates how the 17 features denoted as X_1 until X_{17} being allocated into L_{32} (2^{31}) array structure.

5.3 Threshold value

For this case study, a threshold value of 9 (an MD value) denoted as MD_T is selected based on suggestions by Taguchi and Jugulum (G. Taguchi & Jugulum, 2002) who derived the value based on Taguchi Loss Function computations. Taguchi Loss Function is a decision tool incorporating cost element into the decision rule where it tries to balance or minimize the costs required and the losses incurred in making the decision. Readers can refer to Phadke (Phadke, 1989), Park (Park, 1996) and Taguchi and Jugulum (G. Taguchi & Jugulum, 2002) should one require further readings on Taguchi Loss concepts.

6 RESULTS AND DISCUSSION

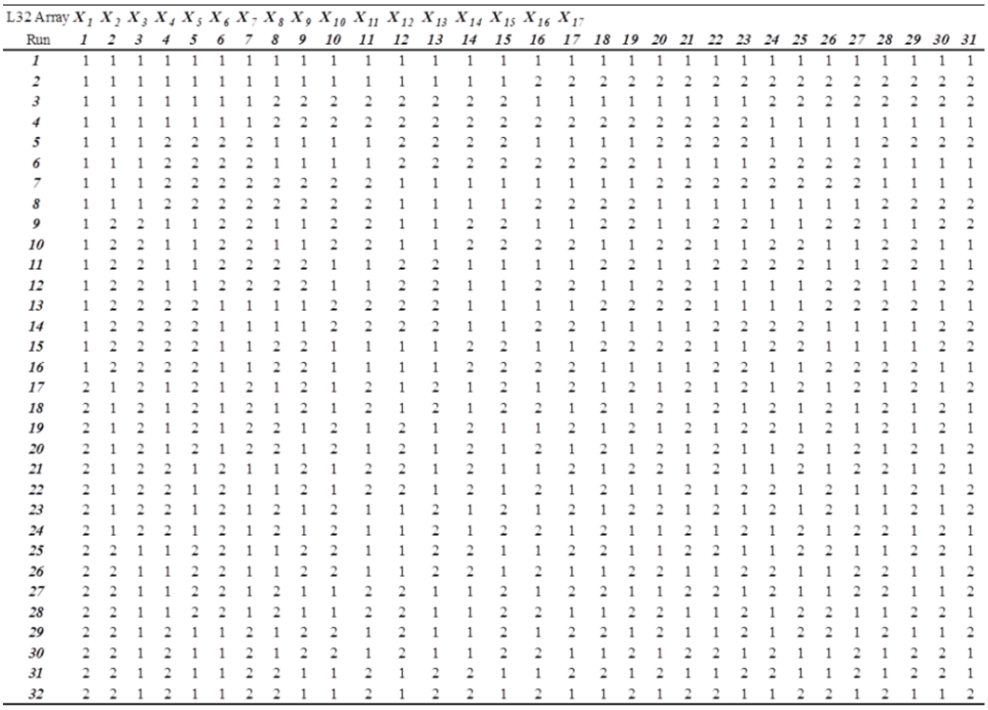
The optimization algorithms for both OA, BA and *mBA* were constructed using Visual Basic language platform. The programming algorithms were then compiled on a 64-bit under high performance computing machine with IntelCore i7-8750H Data Processor; NVIDIA GeForce GTX1060 Image Processor, DDR42666 16GB Memory, GDDR5 6GB Booting Memory, 256GB SSD and 1TB HDD storage capacity.

6.1 Optimization performances

Table 8 shows the SNR gain results on the liver disease data using both OA and *mBA* couples with the optimized features obtained respectively. Note that the exhaustive search results were also tabulated on the same table for benchmarking purpose.

Table 8, *mBA* found a lesser number of optimized features compared to OA with 6 to 8 features, respectively. The types of features optimized by both optimizers were slightly different with only features ‘X_5, X_7, X_{10}, X_{12}, X_{13} and X_{14}’ being considered insignificant by both optimizers. The other insignificant features detected by the OA were ‘X_2’ and ‘X_{15}’. The optimized systems using both OA and *mBA* methods produce higher SNR value as compared to the original set. These suggest that the optimized system using both methods improved the functionality of the system. In addition, the optimized system via *mBA* produced higher SNR gain value as compared to optimized system via OA, with 3.07 dB and 2.46 dB respectively. This suggests that optimizing the system using *mBA* provides greater improvement in terms of overall functionality of the system. In other words, optimizing the system through *mBA* reduces the variability (noise) of the system more in comparison to the improvement done by OA.

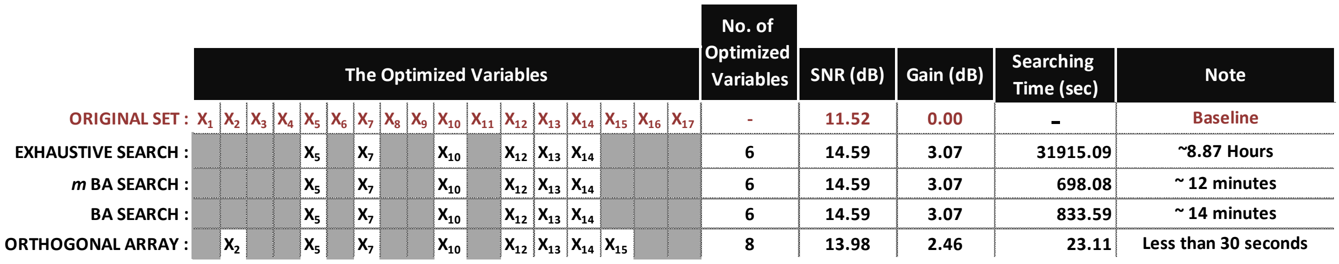
Table 7: Variable allocation in the L₃₂ (2^{31}) orthogonal array



L32 Array	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}														
Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
4	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
5	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2
6	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1
7	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
8	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
9	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
10	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
11	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1
12	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	1	1	2	2	1	1	2	2
13	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1
14	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2
15	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2
16	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2	2	1	1	2	2	2	2	1	1
17	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
18	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
19	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1
20	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	1	2	1	2	1	2	1	2
21	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1
22	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2
23	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2
24	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	1	2	1	2	2	1	2	1
25	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1
26	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2
27	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2
28	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	1	2	2	1	1	2	2	1
29	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2
30	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1
31	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1
32	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2	1	1	2	2	1	2	1	1	2

In finding the optimized features of the liver disease, *mBA* consumed 698.08 seconds. This value is equivalent to around 12 minutes of computational time before the algorithm stopped at the final solution. However, OA took only 23.11 seconds to find the optimum set of features. In other words, OA took just below than a half of a minute before stopping the algorithm at the optimum solution. Thus, OA facilitates the search at a much faster rate.

Table 8: Results on the liver disease optimization performances



	The Optimized Variables																	No. of Optimized Variables	SNR (dB)	Gain (dB)	Searching Time (sec)	Note
ORIGINAL SET :	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	-	11.52	0.00	-	Baseline
EXHAUSTIVE SEARCH :					X_5		X_7			X_{10}		X_{12}	X_{13}	X_{14}				6	14.59	3.07	31915.09	~8.87 Hours
m BA SEARCH :					X_5		X_7			X_{10}		X_{12}	X_{13}	X_{14}				6	14.59	3.07	698.08	~ 12 minutes
BA SEARCH :					X_5		X_7			X_{10}		X_{12}	X_{13}	X_{14}				6	14.59	3.07	833.59	~ 14 minutes
ORTHOGONAL ARRAY :		X_2			X_5		X_7			X_{10}		X_{12}	X_{13}	X_{14}	X_{15}			8	13.98	2.46	23.11	Less than 30 seconds

The reason behind the huge amount of time difference of *mBA* was mainly due to the enormous enumeration set of possible combinations that the 'scout bees' in *mBA* had to search for the optimum feature set. With 17 numbers of original features, the total number of combinations required in the searching space by the 'scout bees' was more than 130 000 different set of combinations. In addition, *mBA* has to search in another enumeration set of possible combinations at the neighbourhood search space. These search processes were to be repeated under several iterations before reaching to the optimal solution.

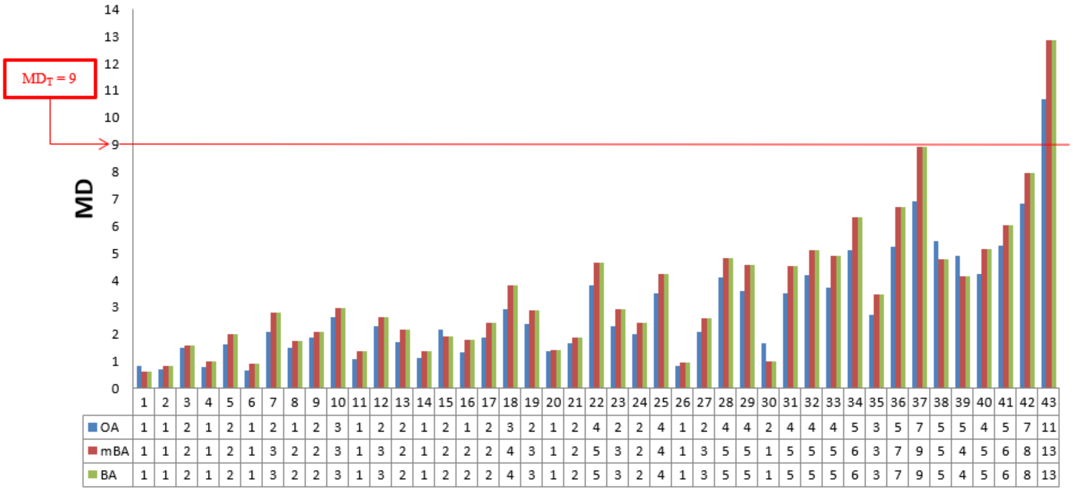
In the contrary, OA search mechanism does not require such search schemes (search in the enumeration of possible combinatorial sets). Furthermore, based on the Latin orthogonal array of L₃₂ (2^{31}) arrangement, only 32 number of iterations (number of runs) were required. Thus under this simple structure, the searching effort was far less as compared to heuristic search burden demonstrated by the 'scout bees' in *mBA*. Consequently, the computational speed was in great faster as compared to *mBA*.

It is worth to mention that the time taken to find the optimum set of liver disease features via exhaustive search was more than 30 000 seconds. This value is equivalent to nearly nine hours period. This is evident (as suggested by (Jain et al., 2000)) that even though exhaustive search is a ‘guaranteed method’ in finding an optimal solution, it is not a viable approach due high computational effort. Of note, the type and number of features optimized through *mBA* were similar to those features optimized via exhaustive search. Thus, in this case, *mBA* was seen to have the ability to find the ‘true optima’ with less searching effort (i.e. faster computing time compared to exhaustive search).

6.2 Classification accuracy of the optimized systems

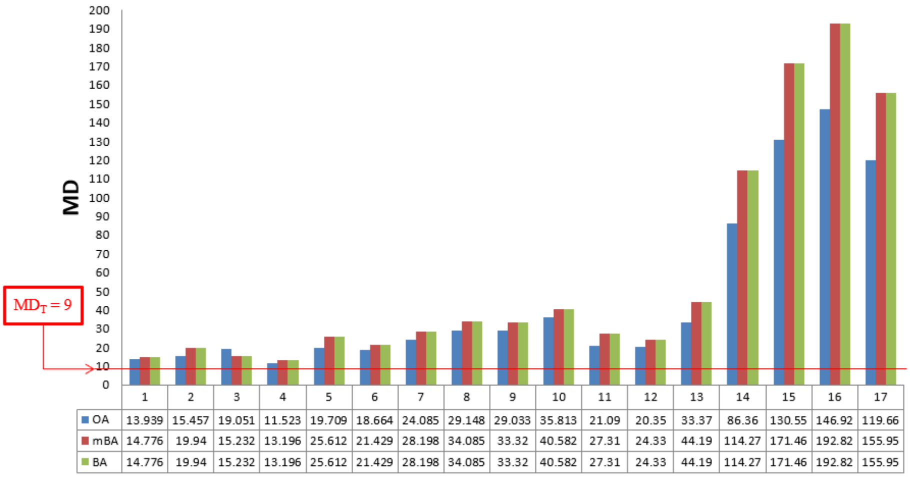
To evaluate the performance of the optimizers under the new sets of optimized features, the system is tested using the testing data set as mentioned in Section 5.1 above. Again, the testing data set contains two types of sample namely healthy group of people (people who do not suffer from liver disease) and unhealthy group. The aim is to calculate the error rate of the optimized systems in classifying correctly the samples to their respective group.

The MD distributions of both samples are illustrated in Figure 12 and Figure 13 respectively. In these figures, the x-axis denotes the series of sample (in numbers) as well as their respective MD values (underneath the numbers). Indication mark of threshold value of $MD_T = 9$ is also portrayed on both figures. Note that the MD data presented on the x-axis Figure 12 are rounded to one decimal place due to space limitation.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
🟦 OA	1	1	2	1	2	1	2	1	2	3	1	2	2	1	2	1	2	3	2	1	2	4	2	2	4	1	2	4	4	2	4	4	4	5	3	5	7	5	5	4	5	7	11
🟥 mBA	1	1	2	1	2	1	3	2	2	3	1	3	2	1	2	2	2	4	3	1	2	5	3	2	4	1	3	5	5	1	5	5	5	6	3	7	9	5	4	5	6	8	13
🟩 BA	1	1	2	1	2	1	3	2	2	3	1	3	2	1	2	2	2	4	3	1	2	5	3	2	4	1	3	5	5	1	5	5	5	6	3	7	9	5	4	5	6	8	13

Figure 12: MD distributions on healthy testing sample using optimized variable sets via OA, BA and *mBA*



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
🟦 OA	13.939	15.457	19.051	11.523	19.709	18.664	24.085	29.148	29.033	35.813	21.09	20.35	33.37	86.36	130.55	146.92	119.66
🟥 mBA	14.776	19.94	15.232	13.196	25.612	21.429	28.198	34.085	33.32	40.582	27.31	24.33	44.19	114.27	171.46	192.82	155.95
🟩 BA	14.776	19.94	15.232	13.196	25.612	21.429	28.198	34.085	33.32	40.582	27.31	24.33	44.19	114.27	171.46	192.82	155.95

Figure 13: MD distributions for unhealthy testing samples using optimized variable sets via OA, BA and *mBA*

From Figure 13, it shows that only one sample (the last sample) was mistakenly classified as unhealthy people by both optimized systems since the MD values obtained were above the threshold value (MD_T=9). In this case, the MD values for this particular sample were computed as 10.676 and 12.838 for OA and *mBA* respectively. In other words, both optimized systems managed to correctly classify 42 out of 43 samples (whom in reality were the healthy people) as healthy people since each of the 42 samples has an MD value less than the threshold value.

On the other hand, both optimized systems managed to classify correctly all the unhealthy samples (17 samples whom in reality were diagnosed as suffering from liver disease) as the correct unhealthy people. There was no error found on both optimized systems when classifying the unhealthy samples. In total, if all the data were to be combined (healthy and unhealthy data are grouped in one dataset), the accuracy rate was calculated to be around 98.33% from each optimizer (refer Table 9).

Table 9: The classification performance of both optimizers

MT System	Classification accuracy rate (%)
Optimized via OA	98.33
Optimized via *mBA*	98.33
Optimized via conventional *BA*	98.33

Table 9 suggests that the optimized system via OA and the optimized system via modified and conventional BAs have an equivalent performance in terms of classification error rate. However, mBA managed to optimize with lesser number of variables in comparison to the OA. Furthermore, mBA converged faster to the optimum solution compared to the original BA which suggests that the proposed architecture of the modified version of the BA expedites the search process in finding the potential solution. Figure 14 supports this argument where it was clearly shown from the figure that *m*BA converged earlier to optimum solution in comparison to BA.

7 CONCLUSION

The ability to classify objects with a smaller number of features to an equal or better recognition accuracy is of major concern in MTS methodology. This paper provides a comparative study to evaluate the effectiveness of OA against swarm intelligence methods to enhance feature selection of the MTS in its optimization scheme.

This paper demonstrates the fusion of a modified-swarm intelligence technique called modified-Bees Algorithm (*mBA*) into the MTS optimization procedure with SNR as its objective function. To the authors' knowledge, the attempts were never been reported in literatures. It reveals a superior computational speed performance of the *m*BA against the conventional BA approach, and with a lesser number of variables in comparison to the OA. However, it is suggested to conduct more case studies with more data to further evaluate the overall effectiveness of the *m*BA in order to generalize the result.

Future research should be focusing on improving the computational speed and exploiting more advantageous of what Bees Algorithm technique could offer to facilitate the feature selection scheme of the MTS with more optimum solution as well as with more efficient search manner and consequently enhancing it recognition performance.

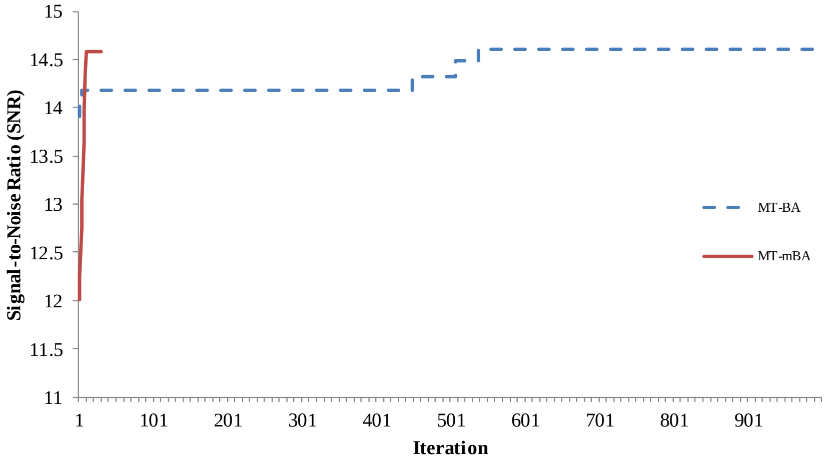


Figure 14: Convergence behavior between the BA and mBA

ACKNOWLEDGEMENT

This work was supported under the Fundamental Research Grant Scheme (FRGS) awarded by the Ministry of Education, Malaysia (Ref: FRGS/1/2019/TK03/UTM/02/10) and Universiti Teknologi Malaysia.

REFERENCES:

[1] Abraham, B., & Variyath, A. M. (2003). Discussion. *Technometrics*, *45*(1), 22–24. https://doi.org/10.1198/004017002188618644

[2] Abu, M. Y., Jamaludin, K. R., & Ramlie, F. (2013). Pattern Recognition Using Mahalanobis-Taguchi System on Connecting Rod through Remanufacturing Process: A Case Study. *Advanced Materials Research*, *845*, 584–589. https://doi.org/10.4028/www.scientific.net/AMR.845.584

[3] Blum, C., & Merkle, D. (2008). *Swarm Intelligence: Introduction and Application* (1st Ed.). Berlin, Heidelberg: Springer-Verlag.

[4] Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization : Overview and Conceptual Comparison. *ACM Computing Surveys*, *35*(3), 268–308.

[5] Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. US: John Wiley and Sons.

[6] Feng, S., Hiroyuki, O., Hidennori, T., Yuichi, K., & Hu, S. (2011). Qualitative and quantitative analysis of gmaw welding fault based on mahalanobis distance. *International Journal of Precision Engineering and Manufacturing*, *12*(6), 949–955. https://doi.org/10.1007/s12541-011-0127-3

[7] Ghasemi, E., Aaghaie, A., & Cudney, E. A. (2015). Mahalanobis Taguchi system: a review. *International Journal of Quality & Reliability Management*, *32*(3), 291–307. https://doi.org/http://dx.doi.org/10.1108/IJQRM-02-2014-0024

[8] Guo, W., Yin, R., Li, G., & Zhao, N. (2013). Research on Selection of Enterprise Management-Control Model Based on Mahalanobis Distance. In E. Qi, J. Shen, & R. Dou (Eds.), *The 19th International Conference on Industrial Engineering and Engineering Management* (pp. 555–564). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38442-4

[9] Hawkins, D. M. (2003). Discussion. *Technometrics*, *45*(1), 25–29. https://doi.org/10.1198/004017002188618653

[10] Hedayat, A. S., Sloane, N. J. A., & Stufken, J. (1999). *Orthogonal Arrays: Theory and Applications* (1st Ed.). New York: Springer-Verlag New York, Inc.

[11] Hwang, I., & Park, G. (2011). A multi-objective optimization using distribution characteristics of reference data for reverse engineering. *International Journal for Numerical Methods in Engineering*, *85*, 1323–1340. https://doi.org/10.1002/nme

[12] Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical Pattern Recognition : A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 4–37.

[13] Krause, J., Cordeiro, J., Parpinelli, R. S., & Lopes, H. S. H. (2013). A Survey of Swarm Algorithms Applied to Discrete Optimization Problems. *Swarm Intelligence and Bio-Inspired Computation*, 169–191. https://doi.org/10.1016/B978-0-12-405163-8.00007-7

[14] Kumar, V., & Minz, S. (2014). Feature Selection: A literature Review. *Smart Computing Review*, *4*(3), 211–229. https://doi.org/10.6029/smartcr.2014.03.007

[15] Mahalanobis, P. C. (1936). On The Generalised Distance in Statistics. *Proceedings of the National Institute of Science of India*, *2*, 49–55.

[16] Muhamad, W. Z. A. W., Jamaludin, K. R., Yahya, Z. R., & Ramlie, F. (2017). A hybrid methodology for the mahalanobis-taguchi system using random binary search-based feature selection. *Far East Journal of Mathematical Sciences*, *101*(12). https://doi.org/10.17654/MS101122663

[17] Muhamad, W. Z. A. W., Ramlie, F., & Jamaludin, K. R. (2017). Mahalanobis-Taguchi system for pattern recognition: A brief review. *Far East Journal of Mathematical Sciences*, *102*(12). https://doi.org/10.17654/MS102123021

[18] Muhamad, W. Z. A. W. W. Z. A. W., Jamaludin, K. R. K. R., Ramlie, F., Harudin, N., & Jaafar, N. N. N. N. (2017). Criteria selection for an MBA programme based on the mahalanobis Taguchi system and the Kanri Distance Calculator. In *2017 IEEE 15th Student Conference on Research and Development (SCOReD)* (Vol. 2018–Janua, pp. 220–223). IEEE. https://doi.org/10.1109/SCORED.2017.8305390

[19] Özbakir, L., Baykasoğlu, A., & Tapkan, P. (2010). Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation*, *215*, 3782–3795. https://doi.org/10.1016/j.amc.2009.11.018

[20] Pal, A., & Maiti, J. (2010). Development of a hybrid methodology for dimensionality reduction in Mahalanobis–Taguchi system using Mahalanobis distance and binary particle swarm optimization. *Expert Systems with Applications*, *37*(2), 1286–1293. https://doi.org/10.1016/j.eswa.2009.06.011

[21] Park, S. H. (1996). *Robust Design and Analysis for Quality Engineering* (1st Ed.). Boundary Row, London: Chapman & Hall.

[22] Phadke, M. S. (1989). *Quality Engineering using Robust Design* (1st Ed.). Englewood Cliffs, New Jersey: Prentice Hall.

[23] Pham, D. T., A. Ghanbarzadeh, Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The Bees Algorithm - A novel tool for complex optimization problems. In *2nd I*PROMS Virtual International Conference: Intelligent Production Machines and Systems* (pp. 454–459). Cardiff, UK: Elsevier Ltd.

[24] Pham, D. T., & Castellani, M. (2009). The Bees Algorithm: modelling foraging behaviour to solve continuous optimization problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *223*(12), 2919–2938. https://doi.org/10.1243/09544062JMES1494

[25] Ramlie, F., Jamaludin, K. R., & Dolah, R. (2016). Optimal Feature Selection of Taguchi Character Recognition in the Mahalanobis-Taguchi System. *Global Journal of Pure and Applied Mathematics*, *12*(3), 2651–2671.

[26] Reséndiz, E., Moncayo-Martínez, L. a., & Solís, G. (2013a). Binary ant colony optimization applied to variable screening in the Mahalanobis–Taguchi System. *Expert Systems with Applications*, *40*(2), 634–637. https://doi.org/10.1016/j.eswa.2012.07.058

[27] Reséndiz, E., Moncayo-Martínez, L. a., & Solís, G. (2013b). Binary ant colony optimization applied to variable screening in the Mahalanobis–Taguchi System. *Expert Systems with Applications*, *40*(2), 634–637. https://doi.org/10.1016/j.eswa.2012.07.058

[28] Reséndiz, E., & Rull-Flores, C. A. (2013). Mahalanobis–Taguchi system applied to variable selection in automotive pedals components using Gompertz binary particle swarm optimization. *Expert Systems with Applications*, *40*(7), 2361–2365. https://doi.org/10.1016/j.eswa.2012.10.049

[29] Reyes-Carlos, Y. I., Mota-Gutiérrez, C. G., & Reséndiz-Flores, E. O. (2018). Optimal variable screening in automobile motor-head machining process using metaheuristic approaches in the Mahalanobis-Taguchi System. *The International Journal of Advanced Manufacturing Technology*, *95*(9–12), 3589–3597. https://doi.org/10.1007/s00170-017-1348-0

[30] Taguchi, G., Chowdhury, S., & Wu, Y. (2001). *The Mahalanobis-Taguchi System* (1st Ed.). New York: McGraw-Hill.

[31] Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's Quality Engineering Handbook*. *John Wiley & Sons, Inc.* (1st Editio). New Jersey: John Wiley & Sons, Inc.

[32] Taguchi, G., & Jugulum, R. (2002). *The Mahalanobis-Taguchi Strategy: A Pattern Technology System* (1st Ed.). New York: John Wiley & Sons, Inc.

[33] Taguchi, G., Rajesh, J., & Taguchi, S. (2004). *Computer-based Robust Engineering: Essentials for DFSS* (1st Ed.). Wisconsin, USA: American Society for Quality.

[34] Taguchi, S., Jugulum, R., Taguchi, G., & Wilkins, J. O. (2003). Discussion. *Technometrics*, *45*(1), 16–21. https://doi.org/10.1198/004017002188618635

[35] Teshima, S., Hasegawa, Y., & Tatebayashi, K. (2012). *Quality Recognition and Prediction: Smarter Pattern Technology with the Mahalanobis-Taguchi System* (1st Ed.). New York: Momentum Press. Retrieved from www.momentumpress.net

[36] Thangavel, K., & Pethalakshmi, a. (2009). Dimensionality reduction based on rough set theory: A review. *Applied Soft Computing*, *9*, 1–12. https://doi.org/10.1016/j.asoc.2008.05.006

[37] Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recogniton* (4th Editio). London, UK: Academic Press.

[38] Tsai, H.-C. (2014). Novel Bees Algorithm: Stochastic self-adaptive neighborhood. *Applied Mathematics and Computation*, *247*, 1161–1172. https://doi.org/10.1016/j.amc.2014.09.079

[39] Woodall, W. H., Koudelik, R., Tsui, K.-L., Kim, S. B., Stoumbos, Z. G., & Carvounis, C. P. (2003). A Review and Analysis of the Mahalanobis—Taguchi System. *Technometrics*, *45*(1), 1–15. https://doi.org/10.1198/004017002188618626

[40] Yang, X.-S. (2010). *Nature-Inspired Metaheuristic Algorithms* (2nd Editio). United Kingdom: Luniver Press.