Applied Soft Computing 20 (2014) 54–65

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hypoglycaemia detection using fuzzy inference system with intelligent optimiser

J.C.Y. Lai[a,*], F.H.F. Leung[a], S.H. Ling[b]

[a] Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong
[b] Centre for Health Technologies, Faculty of Engineering and IT, University of Technology, Sydney, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 3 April 2013
Received in revised form
13 December 2013
Accepted 14 December 2013
Available online 13 January 2014

Keywords:
Differential evolution
Fuzzy inference system
Hypoglycaemia
Multi-objective optimisation

ABSTRACT

Hypoglycaemia is a medical term for a body state with a low level of blood glucose. It is a common and serious side effect of insulin therapy in patients with diabetes. In this paper, we propose a system model to measure physiological parameters continuously to provide hypoglycaemia detection for Type 1 diabetes mellitus (TIDM) patients. The resulting model is a fuzzy inference system (FIS). The heart rate (*HR*), corrected *QT* interval of the electrocardiogram (ECG) signal (QT_c), change of *HR*, and change of QT_c are used as the input of the FIS to detect the hypoglycaemic episodes. An intelligent optimiser is designed to optimise the FIS parameters that govern the membership functions and the fuzzy rules. The intelligent optimiser has an implementation framework that incorporates two wavelet mutated differential evolution optimisers to enhance the training performance. A multi-objective optimisation approach is used to perform the training of the FIS in order to meet the medical standards on sensitivity and specificity. Experiments with real data of 16 children (569 data points) with TIDM are studied in this paper. The data are randomly separated into a training set with 5 patients (l99 data points), a validation set with 5 patients (177 data points) and a testing set with 5 patients (193 data points). Experiment results show that the proposed FIS tuned by the proposed intelligent optimiser can offer good performance of classification.

1. Introduction

In this paper, a fuzzy inference system (FIS) is developed to model the relationship between four physiological parameters and the episodes of hypoglycaemia. Hypoglycaemia is a medical term for a body state with a low level of blood glucose. It is a result of mismatch between the action of insulin, the ingestion of food and the energy expenditure. Hypoglycaemia is not common in non-diabetic persons, but can occur at any age [1]. It can be caused by excessive insulin produced in the body, inborn errors, medications and poisons, alcohol, hormone deficiencies, prolonged starvation, alterations of metabolism associated with infection, and organ failure [2]. Some study reported that diabetic patients, who have been treated with insulin, have high risk of developing hypoglycaemia. Most surveys reported that the tighter the glycaemia control, and the younger the patient, the greater frequency of both mild and severe hypoglycaemia would occur [3,4].

Hypoglycaemia may be defined differently for different people in terms of blood glucose in different circumstances and for different purposes. Most of the healthy adults maintain a fasting glucose level at above 70 mg/dL (3.9 mmol/L), and develop symptoms of hypoglycaemia when the glucose level falls below 55 mg/dL (3.3 mmol/L) [2,5]. Several studies reported that hypoglycaemic episodes are defined as blood glucose levels being below 50 mg/dL (2.8 mmol/L), and the patients are advised to take necessary medical treatment immediately.

The symptoms of hypoglycaemia are often unaware by the patients. Nocturnal hypoglycaemia is particularly dangerous because it may obscure autonomic counter-regulatory responses, so that any initially mild episodes may become severe. 50% of all severe episodes occur at night time. Deficient glucose counter-regulation can lead to severe hypoglycaemia even with the modest insulin elevations. Regulation of nocturnal hypoglycaemia is further complicated by the dawn phenomenon [7]. This is a consequence of nocturnal changes in insulin sensitivity secondary to growth hormone secretion: a decrease in insulin requirement approximately between midnight and 5 am followed by an increase in requirement between 5 am and 8 am. Thus, hypoglycaemia is one of the complications of diabetes most feared by patients. As a result, constructing a model for the detection of hypoglycaemia with respect to some physiological parameters are very important for patients to perform real-time monitoring of their blood glucose level especially at night time [6,8,11,12]. In this paper, a fuzzy

[*] Corresponding author.
E-mail addresses: 08900438r@polyu.edu.hk, johnnylaicy@gmail.com (J.C.Y. Lai), enfrank@inet.polyu.edu.hk (F.H.F. Leung), Steve.Ling@uts.edu.au (S.H. Ling).

1568-4946/$ – see front matter © 2014 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.asoc.2013.12.015

inference system is developed to model the relationship between four physiological signals and the episodes of hypoglycaemia. The physiological signals are the heart rate (HR), the change of HR with time (ΔHR), the corrected QT interval (QT_c) of the electrocardiogram (ECG) signal, and the change of QT_c with time (ΔQT_c). These four signals are used as the inputs for the classification.

In this paper, a fuzzy inference system (FIS) is employed to perform the modelling of the hypoglycaemic episodes. Fuzzy inference is a process of making decisions by using fuzzy logic and fuzzy rules. The data for the hypoglycaemic detection are real-time body signals from patients. A lot of noise and error can be present during the measurement process. Fuzzy logic (FL) can be applied to obtain a good model for processing those body signals. Thanks to the linguistic rule base, any human knowledge can readily be absorbed into the FIS.

Before the FIS works as a classifier for hypoglycaemia episodes for Type 1 diabetes mellitus patients, it has to be trained by some data set with known class labels. The training is a kind of supervised learning. The major objective is to determine the best parameter values for the FIS's rules and membership functions. The training can be realised as an optimisation process. Traditional optimisation methods like the least square algorithm and gradient descent methods have the potential problem of trapping in some local optima of the solution space. Hence, evolutionary optimisation algorithms are considered in this paper.

Differential evolution (DE) has been well accepted as a powerful evolutionary computation algorithm for handling optimisation problems during the last decade. It is a population based stochastic optimisation algorithm that searches the solution space by using the weighted difference between two population vectors to determine a third vector [9,17,18]. Owing to the population-based strategy, DE is less possibly getting trapped in some locally optimal solution. In [22], an improved version of DE with double wavelet mutations (DWM-DE) is employed to the training process of the hypoglycaemia problem. The result shows that DWM-DE performs well in the training process. The details of the performance and comparison after using the wavelet mutation in DE were reported in [25]. However, the performance of DWM-DE still depends quite much on some control parameters' values and the setting of the initial conditions. To reduce these limitations, a further improved intelligent optimiser is introduced, which is designed based on DWM-DE. In the intelligent optimiser, a fuzzy controller is employed to adjust the parameter values adaptively during the progress of searching. To reduce the dependence on the initial conditions of the optimisation algorithm, a parallel implementation framework involving two wavelet-mutated differential evolution (WM-DE) engines is proposed. Moreover, to reduce the number of control parameters as a compensation for the increase of complexity brought by the fuzzy controller, we use a single wavelet operation instead of two. The resulting system consists of two WM-DE engines running in parallel with different initial conditions to tackle the same optimisation problem. The fuzzy controller in the proposed intelligent optimiser captures the on-line population information from the two WM-DE engines. The information is analysed and compared by the Student T-test, which is a method to assess whether two groups of data are statistically different from each other in terms of the mean fitness values. The individual populations' information and the result of the Student T-test act as the inputs of a fuzzy controller to determinate the next iteration (generation) values of the control parameters of the DE engines. The two DE engines in the intelligent optimiser act as a pairing system with additional searching information shared between them. The result is a closed-loop adaptive control system that supports the intelligent optimiser for better performance. Thanks to the changes brought by the intelligent optimiser, the solution reliability can be enhanced when the fuzzy controller tries to minimise the t-value. By applying the wavelet function in the DE's crossover, we can have the solution space to be more widely explored in the early stage of the search; and are more likely to obtain a fine-tuned global solution in the later stage of the search by setting a smaller searching space. The wavelet function's properties enable us to improve the performance of DE in terms of convergence speed, solution quality and solution reliability statistically [22].

In this paper, sensitivity and specificity are used to measure the performance of the classification. The sensitivity measures the proportion of actual positives that are correctly identified; and the specificity measures the proportion of actual negatives that are correctly identified. As common clinical classification requirements, the sensitivity should be higher than 70% and the specificity should be higher than 50% in order to obtain a reliable classification. Based on these clinical requirements, we have two objectives to achieve in the training process. As a result, a multi-objective optimisation approach using the proposed intelligent optimiser should be employed to realise the training process. Besides, overtraining is another key problem that affects the classification performance when constructing the FIS model. Overtraining refers to the reduction of the generalisation ability that can occur as a system is trained. In this paper, a validation strategy is employed to reduce the risk of overtraining [20,21,23]. This strategy is embedded in the training operation and the formulation of the fitness functions in the proposed intelligent optimiser. The details will be given in Section 2.

The organisation of this paper is as follows: in Section 2, the details of the development of the FIS and the proposed intelligent optimiser are presented. The experiment results of detecting nocturnal hypoglycaemic episodes in TIDM patients are discussed in Section 3. A conclusion is drawn in Section 4.

2. Fuzzy inference system with the proposed intelligent optimiser

To realise the detection of hypoglycaemic episodes for the type-1 diabetes mellitus (T1DM) cases, an FIS tuned by the proposed intelligent optimiser with two wavelet-mutated differential evolution (WM-DE) engines is employed. A block diagram of the FIS is shown in Fig. 1. It is a 4-input and 1-output learning and modelling system. The physiological inputs are the heart rate (HR), the corrected QT interval of the electrocardiogram signal (QT_c), the change of heart rate (ΔHR), and the change of QT_c (ΔQT_c). The output is the binary state of hypoglycaemia (h), which takes the value of true (+1) or false (−1). The major role of the FIS is to model the relationship of HR, QT_c, ΔHR and ΔQT_c to h so as to perform classification.

Two of the inputs of the FIS are obtained from the ECG signal. The ECG signal being investigated involves the parameters in the depolarisation and repolarisation stages of electrocardiography. An example ECG signal (with two cycles) is shown in Fig. 2. In this study, the concerned points are the Q points, R peak, T wave peak (T_p), and the T wave end (T_e). The QT interval is between the Q point

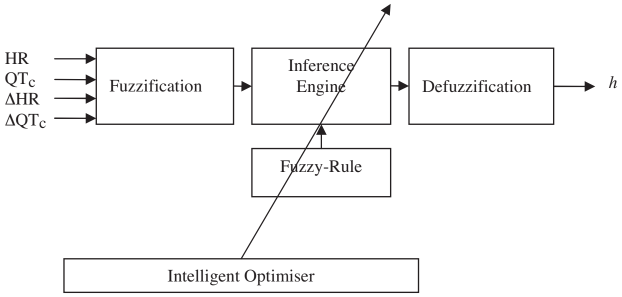


Fig. 1. Fuzzy inference system (FIS).

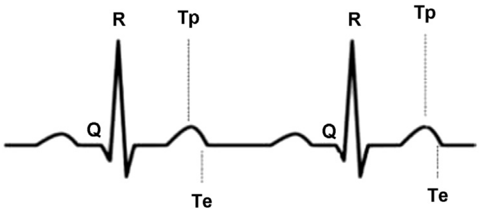


Fig. 2. ECG signal.

and the T wave peak. The QT_c signal is obtained by QT/RR, where RR is the interval between two consecutive R peaks. The heart rate is given by $60/RR$.

2.1. Fuzzy inference system

In the classifier, the physiological data (HR, QT_c, ΔHR and ΔQT_c) are represented as linguistic variables governed by membership functions that support fuzzy descriptions. The FIS contains four major parts: fuzzification, inference engine, fuzzy-rule base, and defuzzification.

On doing fuzzification, the inputs are mapped to different membership functions by using fuzzy sets. Bell-shaped fuzzy membership functions are used and defined as follows:

$$f_m(x(t), \sigma_m, c_m)e^{(xc_m)^2/(2m^2)} \tag{1}$$

where $x(t)$ is the non-fuzzy input, $m = 1, 2, \ldots, m_f$; m_f denotes the number of membership functions; σ_m and c_m are the standard deviation and the mean value of the membership function respectively. Fig. 14 shows a set of bell-shaped fuzzy membership functions for an input variable. It contains three linguistic terms (M1, M2, and M3). The input range is between 0 and 1. In the FIS, all the inputs are mapped to the corresponding fuzzy membership functions. The membership functions for the four inputs are given by:

$$f_{HR,m}(HR(t), \sigma_{HR,m}, c_{HR,m}) = e^{-(HR(t)-c_{HR,m})^2/(2(\sigma_{HR,m})^2)} \tag{2}$$

$$f_{QT_c,m}(QT_c(t), \sigma_{QT_c,m}, c_{QT_c,m}) = e^{-(QT_c(t)-c_{QT_c,m})^2/(2(\sigma_{QT_c,m})^2)} \tag{3}$$

$$f_{\Delta HR,m}(\Delta HR(t), \sigma_{\Delta HR,m}, c_{\Delta HR,m}) = e^{-(\Delta HR(t)-c_{\Delta HR,m})^2/(2(\sigma_{\Delta HR,m})^2)} \tag{4}$$

$$\begin{aligned} &f_{\Delta QT_c,m}(\Delta QT_c(t), \sigma_{\Delta QT_c,m}, c_{\Delta QT_c,m}) \\ &\quad = e^{-(\Delta QT_c(t)-c_{\Delta QT_c,m})^2/(2(\sigma_{\Delta QT_c,m})^2)} \end{aligned} \tag{5}$$

In the inference engine of the FIS, the fuzzified inputs are processed with respect to a rule base, which generates the fuzzified output. The rule base contains a set of fuzzy if–then rules of the following format:where τ is the rule number, $\tau = 1, 2, \ldots, n_r$; n_r is the number of rules in the fuzzy rule base. The value of n_r is defined by the following equation:

Rule τ.	**IF**	$HR(t)$	is	$f_{HR,m}(HR(t), \sigma_{HR,m}, c_{HR,m})$
	AND	$QT_c(t)$	is	$f_{QTc,m}(QT_c(t), \sigma_{QTc,m}, c_{QTc,m})$
	AND	$\Delta HR(t)$	is	$f_{\Delta HR,m}(\Delta HR(t), \sigma_{\Delta HR,m}, c_{\Delta HR,m})$
	AND	$\Delta QT_c(t)$	is	$f_{\Delta QTc,m}(\Delta QT_c(t), \sigma_{\Delta QTc,m}, c_{\Delta QTc,m})$,
	THEN	$y(t)$	is	w_τ

$$n_r = (m_f)^n \tag{6}$$

where $n = 4$ is the number of inputs of the FIS. After fuzzifying the input signals, aggregation is then realised to each rule to represent the output as a single fuzzy set. The aggregation output for each rule is defined as follows:

```
begin
Initialise the population
while (not termination condition) do
  begin
      Mutation operation by equation (11)
      Crossover operation by equation (12)
      Evaluation of the fitness function
      Select the best vector by equation (13)
  end
end
```

Fig. 3. Pseudo code for SDE.

$$\begin{aligned} o_\tau &= f_{HR,m}(HR(t), \sigma_{HR,m}, c_{HR,m}) \times f_{QT_c,m}(QT_c(t), \sigma_{QT_c,m}, c_{QT_c,m}) \\ &\quad \times f_{\Delta HR,m}(\Delta HR(t), \sigma_{\Delta HR,m}, c_{\Delta HR,m}) \\ &\quad \times f_{\Delta QT_c,m}(\Delta QT_c(t), \sigma_{\Delta QT_c,m}, c_{\Delta QT_c,m}) \end{aligned} \tag{7}$$

During the inference process, we map the fuzzy set outputs of the rules to a crisp output by performing a process called defuzzification. In this thesis, the defuzzification is realised as:

$$y(t)\frac{\sum_{\tau=1}^{n_r} O_\tau w_\tau}{\sum_{\tau=1}^{n_r} O_\tau} \tag{8}$$

where $w_\tau \in [-1, 1]$, $\tau = 1, 2, \ldots, n_r$ are fuzzy singletons to be determined in the training process. If the output $y(t)$ is positive, it means that hypoglycaemia is present. The actual output of the FIS is realised as:

$$h(t) = \begin{cases} +1 & \text{if } y(t) \geq 0. \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

As a result, the output value of the classifier is binary.

2.2. Intelligent optimiser with wavelet-mutated differential evolution

2.2.1. Differential evolution (DE)

To implement the optimisation process using differential evolution, a population in the solution space should be randomly generated. The population of solution vectors are then successively updated and swapped until the population converges to the optimum within the solution space. The pseudo code for the standard DE (SDE) process is shown in Fig. 3. In this paper, a wavelet-mutated DE (WM-DE) algorithm is employed in the intelligent optimiser. The pseudo code of WM-DE is shown in Fig. 4. The details of both SDE and WM-DE are given as follows:

```
begin
Initialise the population
while (not termination condition) do
    begin
        Mutation operation by equation (11)
        Crossover operation by equation (12)
        Modifying the trial population vectors based on
        equation (18)
        Evaluation of the fitness function
        Select the best vector by equation (13)
    end
end
```

Fig. 4. Pseudo code for WM-DE.

```
begin
Initialise the population
While (not termination condition) do
    begin
        Mutation operation by equation (11)
        Crossover operation by equation (12)
        Modifying the trail population vectors based on
        equation (18)
        Evaluation of the fitness functions
        Select the best vector by equation (28)
    end
end
```

Fig. 5. Pseudo code for multi-objective WM-DE.

2.2.2. Standard differential evolution (SDE)

DE attempts to maintain a population of N_p vectors for each generation of evolution, with each vector contains D elements. Let $P_{x,g}$ be the population of the current generation g, and $x_{i,g}$ be the ith vector in this population:

$$\begin{aligned} P_{x,g} &= (\mathbf{x}_{i,g}), \quad i = 0, 1, \ldots, N_p - 1; \quad g = 0, 1, \ldots, g_{\max} \\ P_{i,g} &= (\mathbf{x}_{j,i,g}), \quad j = 0, 1, \ldots, D - 1. \end{aligned} \tag{10}$$

where $g_{\max}$ is the maximum generation number. On doing optimisation, DE creates a mutated vector $\mathbf{v}_{i,g}$ for each target vector $\mathbf{x}_{i,g}$ by using the mutation operation. The mutated vector is therefore realised by the following equation:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \tag{11}$$

where F is the scaling factor; r_1 and r_2 are two different integers, which are randomly generated from $\{0, 1, \ldots, N_p - 1\}$. To complement the mutation search strategy and increase the diversity of the perturbed vectors, DE employs a method called uniform crossover for all the mutated vectors. The crossover operation is realised by the following equation:

$$u_{i,g} = (u_{j,i,g}) = \begin{cases} (v_{j,i,g}) & \text{if } (rand_j(0,1)) \le C_r \\ x_{j,i,g} & \text{otherwise} \end{cases} \tag{12}$$

where $C_r \in [0, 1]$ is called the crossover rate, which is a user-defined value that controls the fraction of parameters copied from the mutant. $rand_j(0,1)$ generates a random value between 0 and 1 for the jth parameter. The algorithm also ensures $u_{j,i,g}$ gets at least one value as $x_{j,i,g}$. Then the population is updated by comparing each trial vector $\mathbf{u}_{i,g}$ to the corresponding target vector $\mathbf{x}_{i,g}$. If the fitness function value of the trial vector is smaller than that of the target vector, replace the target vector in the next generation; otherwise the target vector retains its place in the population for another generation. The selection operation is therefore realised by the following equation:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) \le f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases} \tag{13}$$

where $f(\cdot)$ is the fitness function. Because of this selection operation, DE is expected to have high optimisation ability. When the condition to stop further evolution is satisfied; for example, a preset maximum number of iteration has been reached, the algorithm ends with the best solution as the final solution.

```
begin
Initialise the target value
$\xi_{target}$=0.10
$\eta_{target}$= 0.10
While (not termination condition) do
   begin
       Output ($\xi_{train}$,$\xi_{val}$) by Equation (29) and WM-DE
       Output ($\eta_{train}$,$\eta_{val}$) by Equation (30) and WM-DE
       Output ($f_1$) by Equation (31)
       Output ($f_2$) by Equation (32)
       Output ($f_3$) by Equation (33)
       Output ($f_4$) by Equation (34)
       if($f_1$=$\xi_{target}$& $f_3$=$\xi_{target}$)
            then$\xi_{target}$=$\xi_{target}$+ 0.01
       if($f_2$=$\eta_{target}$& $f_4$=$\eta_{target}$)
            then$\eta_{target}$ =$\eta_{target}$ + 0.01
       end
   end
end
```

Fig. 6. Pseudo code for training and validation.

2.2.3. Wavelet-mutated differential evolution

In the DE crossover operation, we proposed a wavelet operation that varies the searching based on the wavelet function. As the wavelet function output is set to be inversely proportional to the number of iteration; when the searching population is approaching the optimal solution, the effect of the wavelet mutation decreases until the DE ends eventually. By adopting this method, the effort on searching and evaluating those local optima, which could be far away from the global optimum, in the later iteration is reduced. The total number of iteration to obtain the best solution should also decrease. Thanks to the property of the wavelet function, the solution reliability is enhanced in a statistical sense, i.e. the performance of this hybrid DE on converging to the optimal point is relatively stable despite the presence of many random factors during the evolution in DE.

2.2.3.1. Wavelet function. The proposed wavelet operation is based on the wavelet function. In Mathematics, certain seismic signals can be modelled by combining translations and dilations of an oscillatory function called a “wavelet”. A continuous-time function $\psi(x)$ is called a “mother wavelet” or “wavelet” if it satisfies the following properties:

Property 1.

$$\int_{-\infty}^{+\infty} \psi(x)dx = 0 \tag{14}$$

In other words, the total positive momentum of $\psi(x)$ is equal to the total negative momentum of $\psi(x)$.

Property 2.

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 dx < \infty \tag{15}$$

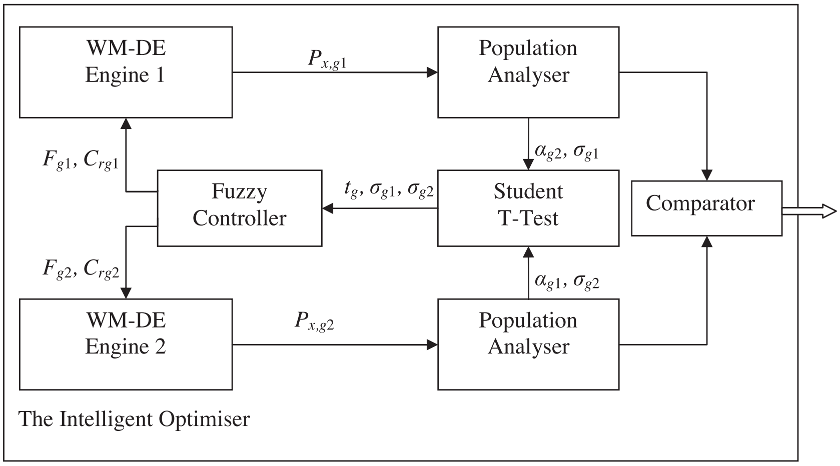
which means most of the energy in $\psi(x)$ is confined to a finite duration and bounded. The Morlet wavelet [16] is an example mother wavelet:

$$\psi(x) = e^{-x^2/2}\cos(5x) \tag{16}$$

The Morlet wavelet integrates to zero (Property 1). Over 99% of the total energy of the function is contained in the interval of $-2.5 < x < 2.5$ (Property 2). In order to control the magnitude of $\psi(x)$, a function $\psi_a(x)$ is defined as follows.

$$\psi_a(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x}{a}\right) \tag{17}$$

where a is the dilation parameter. It follows that $\psi_a(x)$ is an amplitude-scaled version of $\psi(x)$. The amplitude of $\psi_a(x)$ will be



$P_{x,g1}$:	Current population from WM-DE Engine 1
$P_{x,g2}$:	Current population from WM-DE Engine 2
F_{g1}:	Next F for WE-DE Engine 1
F_{g2}:	Next F for WE-DE Engine 2
C_{rg1}:	Next C_r for WE-DE Engine 1
C_{rg2}:	Next C_r for WE-DE Engine 2
α_{g1}:	Mean of fitness from WE-DE Engine 1's population
α_{g2}:	Mean of fitness from WE-DE Engine 2's population
σ_{g1}:	Standard deviation of fitness from WE-DE Engine 1's population
σ_{g2}:	Standard deviation of fitness from WE-DE Engine 2's population
t_g:	Student T-test result based on populations of WE-DE engines

Fig. 7. Block diagram of intelligent optimiser.

scaled down as the dilation parameter a increases. This property is used to do the mutation operation in order to enhance the searching performance.

2.2.3.2. Operation of DE crossover with wavelet mutation. The crossover operation of (12) is done with respect to the elements of the trial vector in DE. In WM-DE, a wavelet mutation is embedded in the crossover operation. It is realised by adding a wavelet mutation following the original crossover operation. The crossover after the first mutation takes place according to (12). Let $u_{i,g} = (u_{0,i,g}, u_{1,i,g}, \ldots, u_{D-1,i,g})$ (where g is the current generation number and D is the number of elements in the vector) be the ith vector after crossover for the wavelet mutation. The value of the element $u_{j,i,g}$ is inside the vector element's boundary $[para^j_{\min}, para^j_{\max}]$. The mutated crossover vector is given by $\bar{\mathbf{u}}_{i,g} = (\bar{u}_{0,i,g}, \bar{u}_{1,i,g}, \ldots, \bar{u}_{D-1,i,g})$, and

$$\bar{u}_{j,i,g} = \begin{cases} u_{j,i,g} + \sigma \times (para^j_{\max} - u_{j,i,g}) & \text{if } \sigma > 0 \\ u_{j,i,g} + \sigma \times (u_{j,i,g} - para^j_{\min}) & \text{if } \sigma \le 0 \end{cases}, \tag{18}$$

$$\sigma = \frac{1}{\sqrt{a}} e^{-(\varphi/a)^2/2} \cos\left(5\left(\frac{\varphi}{a}\right)\right)$$

$$a = e^{-\ln(\lambda)\times\left(1-(t/T)\right)^{\zeta_{wm}} + \ln(\lambda)} \tag{19}$$

where ζ_{wm} is the shape parameter of the monotonic increasing function, λ is the upper limit of the value of parameter a. The Morlet wavelet in (16) is used as the mother wavelet. As a result, a larger value of $|\sigma|$ at the early stage of evolution gives a larger searching space for the solution; when $|\sigma|$ is small at the later stage of evolution, the algorithm gives a smaller searching space for fine-tuning. The details on determining the value of a are discussed in [25]. After the operations of the wavelet mutation, the population is updated by comparing each trial vector $\bar{\mathbf{u}}_{i,g}$ to the corresponding target vector $\mathbf{x}_{i,g}$ using the method of standard DE as given by (13). A new population is generated and the same evolution process is repeated. Such an iterative process will be terminated when a defined number of iteration has been reached. Fig. 4 shows the pseudo code of WM-DE. The details of the performance and comparison after using wavelet mutation in DE were reported in [25].

2.2.4. Intelligent optimiser

The proposed intelligent optimiser has an implementation framework that contains several elements. A block diagram of it is shown in Fig. 7. In addition, the pseudo code for its implementation is given in Fig. 9. The major elements in the intelligent optimiser are two WM-DE engines. On doing the optimisation, the two WM-DE engines run in parallel to tackle the same optimisation problem with two different initial populations. The initial populations are generated randomly at the beginning of the searching process. When the two WM-DE engines are operating, their individual populations at each generation (iteration) will be analysed by the population analyzers as shown in Fig. 7.

In the population analyzer, the fitness value of each vector is calculated. Then, the mean and standard deviation of fitness values within the population of each WM-DE engine are found. The Student T-test algorithm analyses the difference between the two populations and generates a t-value. The t-value and the standard deviations obtained from the two populations are input to the internal fuzzy controller, which determines the F and C_r values for the WM-DE engines for the next iteration (generation). As a result, a closed-loop adaptive intelligent optimiser is realised. At the end of the evolution, a comparator is employed to compare the results from the two WM-DE engines and adopt the best solution as the finally result of the intelligent optimiser. The detailed implementation of the intelligent optimiser is given as follows.

2.2.4.1. Population analyser. The major objective of the population analyser is to capture the movement of the population. In each iteration, the populations of the two WM-DE engines update themselves within their individual searching domains. In principle, after a number of iteration, all the population within the same WM-DE engine should move to the same region where the global optimal point is located. As a result, by monitoring the mean and standard deviation of the fitness values of the vectors in the population, we could acquire information about the progress of searching of the WM-DE engine. Along the searching process, the standard deviation should be gradually decreasing until it reaches a very small value. In the proposed system, two WM-DE engines are employed to tackle the same objective function. We could assume that the means of the two WM-DE engines have similar values in the later stage. Both the mean and the standard deviation are thus used to analyse the progress of searching.

The normalised mean fitness value of the current generation within a WM-DE engine is given by:

$$\alpha = \frac{1}{N_p}\sum_{i}^{N_p}\left\|f(x_{i,g})\right\| \tag{20}$$

where N_p is the number of vectors in the current generation, $\|\cdot\|$ denotes the l_2 norm, and $x_{i,g}$ is one of the vectors in the current generation. The normalised standard deviation of the fitness value of the current generation within a WM-DE engine is given by:

$$\sigma_g = \sqrt{\frac{1}{N_p}\sum_{i}^{N_p}(||f(x_{i,g})|| - \alpha_g)^2} \tag{21}$$

2.2.4.2. Student T-test. In the proposed intelligent optimiser, the Student *T*-test is employed to identify the population difference between the two WM-DE engines. The Student *T*-test is a useful method to perform Hypothesis Test (HT) in the field of statistical research. HT works by collecting data and measuring the difference between the particular sets of data to prove the null hypothesis, which determines an initial guess of some experimental result. If the initial guess of the result is correct, we label the null hypothesis as TRUE, otherwise as FALSE. The *t*-value generated by the Student *T*-test is a ratio between the means difference and standard deviations difference of the two sample groups:

$$t_g = \frac{\overline{\alpha_{g2}} - \alpha_{g1}}{\sqrt{(\sigma_{g2}^2/\xi_{degree}) + (\sigma_{g1}^2/\xi_{degree})}} \tag{22}$$

where t_g is the *t*-value of the gth generation; $\overline{\alpha_{g1}}$ and $\overline{\alpha_{g2}}$ are the mean values from the first and second engines respectively; σ_{g1} and σ_{g2} are the standard deviations from the first and second engines respectively; and ξ_{degree} is the number of samples in the groups minus 2.

The *t*-value can be zero, positive and negative. If the *t*-value is zero, it means that there is no difference between the two groups. To make the *t*-value meaningful, we need to consider the significance of the HT to assure the result is statistically meaningful. To understand the significance, we need to set an alpha level, which is generally considered as a risk level. In most of the scientific research, we set the alpha level at 0.05. The alpha level governs the probability that the means of the two tested groups have a statistically significant difference. The significance of *t*-value changes with the degree of freedom ξ_{degree}. Given the alpha level, the degree of freedom ξ_{degree}, and the *t*-value, we can determine whether the *t*-value is significant enough to prove the null hypothesis.

In the proposed optimiser, the value of ξ_{degree} is determined by the following equation:

$$\xi_{degree} = N_{p1} + N_{p2} - 2 \tag{23}$$

where N_{p1} is the number of vectors in the population of the WM-DE engine 1 and N_{p2} is the number of vectors in the population of the WM-DE engine 2. When we are using the intelligent optimiser, we have a null hypothesis that there should not have any significant difference between the two WM-DE engines as they are solving the same problem. Along the searching operation, the *t*-value should be kept small and less than the significant level. However, in most of the cases during the evolution, the *t*-value could not be kept small because of the randomness in the searching process of WM-DE. Therefore, an internal fuzzy controller is employed in the proposed optimiser to control the searching process of WM-DE in order to reduce the chance that the two populations have significant differences.

2.2.4.3. Fuzzy controller. The fuzzy controller performs fuzzy inference, which is a process of making decisions by using fuzzy logic and fuzzy rules that involves membership functions and logical operations. The membership functions enable the description of the inputs and outputs in linguistic terms. The logical operations and fuzzy rules can be derived based on human knowledge.

The Mamdani method is used to implement the fuzzy inference system to determine the WM-DE parameters. It uses linguistic control rules obtained from experienced human experts to realise making decision. The output variables are fuzzy sets. To map a fuzzy set to some crisp output value, defuzzification is performed, which is a process of generating quantifiable results from fuzzy sets and the corresponding membership functions. In the proposed intelligent optimiser, the method of centre of gravity (COG) is used to perform defuzzification.

WM-DE is a population-based searching algorithm. The population along the generations will not be the same for different experiments of a given problem because the algorithm itself is stochastic. On analysing the population distribution in the searching process, the way of choosing the parameter values of the WM-DE engine is described as some heuristic rules. The fuzzy controller then determines the *F* and C_r values for the next iteration. The fuzzy controller takes the *t*-value of the current generation and the standard deviations of the population fitness from the two WM-DE engines as inputs, and maps the information into the fuzzy membership functions. A closed-loop control system is formed in the proposed intelligent optimiser such that the parameter values of the two WM-DE engines can vary with the searching states.

A fuzzy controller processes linguistic variables, and each variable normally consists of a set of linguistic terms. In the proposed intelligent optimiser, the linguistic terms are modelled by triangular-shaped membership functions, which can be written as:

$$f_m(x_m(t), a_m, b_m, c_m) = \begin{cases} 0, & x_m(t) \le a_m \\ \dfrac{x_m(t) - a}{b_m - a_m} & a_m \le x_m(t) \le b_m \\ \dfrac{c_m - x_m(t)}{c_m - b_m} & b_m \le x_m(t) \le c_m \\ 0, & c_m \le x_m(t) \end{cases} \tag{24}$$

where $x_m(t)$ is the input, $m = 1, 2, \ldots, m_f$; m_f denotes the number of membership functions. Fig. 13 represents three triangular-shaped membership functions (of 3 fuzzy terms M1, M2 and M3) for an input variable. The input range is between zero and one. Every input and output of the fuzzy controller in the proposed intelligent optimiser has three linguistic terms that cover different values of input and output. The reason for choosing the triangular-shaped membership functions is that it is less computational demanding than many other types of membership functions. Only simple arithmetic operations are involved.

In the proposed system, the membership functions for the three inputs are $f_{tg,m}(t_g, a_{tg,m}, b_{tg,m}, c_{tg,m})$, $f_{\sigma g1,m}(\sigma_{g1}, a_{\sigma g1,m}, b_{\sigma g1,m}, c_{\sigma g1,m})$ and $f_{\sigma g2,m}(\sigma_{g2}, a_{\sigma g2,m}, b_{\sigma g2,m}, c_{\sigma g2,m})$. And the four outputs are $f_{Fg1,m}(F_{g1}, a_{Fg1,m}, b_{Fg1,m}, c_{Fg1,m})$, $f_{Fg2,m}(F_{g2}, a_{Fg2,m}, b_{Fg2,m}, c_{Fg2,m})$, $f_{Crg1,m}(C_{rg1}, a_{Crg1,m}, b_{Crg1,m}, c_{Crg1,m})$ and $f_{Crg2,m}(C_{rg2}, a_{Crg2,m}, b_{Crg2,m}, c_{Crg2,m})$. In the inference process, the fuzzified inputs are processed with respect to a rule base, which generates the fuzzified output. The rule base contains a set of fuzzy if-then rules of the following four formats, one for each output:where τ is the rule number, $\tau = 1, 2, \ldots, n_r$; n_r is the number of rules in the fuzzy rule base.

Rule τ:	**IF**	t_g	is	$f_{tg,m}(t_g, a_{tg,m}, b_{tg,m}, c_{tg,m})$
	AND	σ_{g1}	is	$f_{\sigma g1,m}(\sigma_{g1}, a_{\sigma g1,m}, b_{\sigma g1,m}, c_{\sigma g1,m})$
	THEN	F_{g1}	is	$f_{Fg1,m}(F_{g1}, a_{Fg1,m}, b_{Fg1,m}, c_{Fg1,m})$
Rule τ:	**IF**	t_g	is	$f_{tg,m}(t_g, a_{tg,m}, b_{tg,m}, c_{tg,m})$
	AND	σ_{g2}	is	$f_{\sigma g2,m}(\sigma_{g2}, a_{\sigma g2,m}, b_{\sigma g2,m}, c_{\sigma g2,m})$
	THEN	F_{g2}	is	$f_{Fg2,m}(F_{g2}, a_{Fg2,m}, b_{Fg2,m}, c_{Fg2,m})$
Rule τ:	**IF**	t_g	is	$f_{tg,m}(t_g, a_{tg,m}, b_{tg,m}, c_{tg,m})$
	AND	σ_{g1}	is	$f_{\sigma g1,m}(\sigma_{g1}, a_{\sigma g1,m}, b_{\sigma g1,m}, c_{\sigma g1,m})$
	THEN	C_{rg1}	is	$f_{Crg1,m}(C_{rg1}, a_{Crg1,m}, b_{Crg1,m}, c_{Crg1,m})$
Rule τ:	**IF**	t_g	is	$f_{tg,m}(t_g, a_{tg,m}, b_{tg,m}, c_{tg,m})$
	AND	σ_{g2}	is	$f_{\sigma g2,m}(\sigma_{g2}, a_{\sigma g2,m}, b_{\sigma g2,m}, c_{\sigma g2,m})$
	THEN	C_{rg2}	is	$f_{Crg2,m}(C_{rg2}, a_{Crg2,m}, b_{Crg2,m}, c_{Crg2,m})$

After fuzzifying the input signals, aggregation is then realised to each rule to represent the output as a single fuzzy set. The aggregation output for each rule is defined as follows:

$$O_\tau = \min(IN_{1,\tau}, IN_{2,\tau}) \tag{25}$$

where $IN_{1,\tau}$ and $IN_{2,\tau}$ are the two input membership function value of the fuzzy rule τ. The fuzzy output of each rule is realised as follows.

$$OUT_\tau = O_\tau \cdot OUTPUT_\tau \tag{26}$$

where OUT_τ is the output membership function of the corresponding rule and $OUTPUT_\tau$ is the defined output membership function.

During the inference process, we map all the output membership functions from all rules to form the aggregated result for performing defuzzification. The process of defuzzification is used to determinate the numerical output of the fuzzy system. In the intelligent optimiser, the defuzzification is implemented by using the Centroid method, which uses the centroid point of the aggregated result as the numerical result. The Centroid defuzzification is given by,

$$y(t) = \frac{\int OUT_{all} \cdot y dy}{\int OUT_{all} dy} \tag{27}$$

where OUT_{all} is the aggregated result of all the corresponding rules, and $y(t)$ is the corresponding numerical output of the fuzzy system.

In the proposed intelligent optimiser, the membership functions for the inputs and outputs of the internal fuzzy controller are shown in Figs. 10 and 11. The design of the membership functions is based on many experiments with the knowledge of the characteristics of DE. For the input of normalised standard deviation, the range is between 0 and 0.2. We consider the value smaller than 0.05 as LOW and the value larger than 0.15 as HIGH. For the input of t-value, the range is between 0 and 10. When the t-value is higher than 1.645 (with $\xi = 50$), there is a significant difference between the two engines' populations with a 95% confidence level. As a result, we consider the t-value smaller than 1.645 as LOW.

For the outputs of F and C_r, they range between 0 and 0.9. In most of the application, DE works well with the values of F and C_r between 0.3 and 0.5. We consider the value of F and C_r as LOW if they are smaller than 0.1, and we consider the value of F and C_r as HIGH if they are larger than 0.7.

In the proposed intelligent optimiser, it consists of 36 rules. They areThe first of 18 rules are used for the F parameter and the second 18 rules are used for the C_r parameter. In SDE, the F parameter is very important for the mutation operation that produces a new vector for the next generation. On doing the DE mutation operation, the difference of two vectors is calculated and then multiplied by F. Then, it is added to one of the vectors in the population to obtain a new vector for the next generation. The F parameter controls the displacement from the old vector to the new vector. A large value of F means the new vector will move far away from the original position. The basic design principle of the fuzzy rules is that we would like to have the solution space to be more widely explored in the early stage of the search; and it is more likely to obtain a fine-tuned global solution by setting a smaller searching space in the later stage of the search. By examining the population standard deviation of the fitness values, we can estimate the progress of searching of the WM-DE engine. In practice, the population is randomly distributed in the solution space in the early stage of the searching process, making the standard deviation to be large and the value of F to be high. It moves towards the global optimal point along the searching at the later stage, making the standard deviation to be small and the value of F to be low.

For the *F* parameter:

1.	IF (σ_{g1} is low)	AND (t_g is low)	THEN (F_{g1} is low)
2.	IF ($\sigma_{g1}$is medium)	AND (t_g is low)	THEN (F_{g1} is low)
3.	IF (σ_{g1} is high)	AND (t_g is low)	THEN (F_{g1} is medium)
4.	IF (σ_{g1} is low)	AND (t_g is medium)	THEN (F_{g1} is medium)
5.	IF (σ_{g1} is medium)	AND (t_g is medium)	THEN (F_{g1} is medium)
6.	IF (σ_{g1} is high)	AND (t_g is medium)	THEN (F_{g1} is high)
7.	IF (σ_{g1} is low)	AND (t_g is high)	THEN (F_{g1} is medium)
8.	IF (σ_{g1} is medium)	AND (t_g is high)	THEN (F_{g1} is medium)
9.	IF (σ_{g1} is high)	AND (t_g is high)	THEN (F_{g1} is high)
10.	IF (σ_{g2} is low)	AND (t_g is low)	THEN (F_{g2} is low)
11.	IF (σ_{g2} is medium)	AND (t_g is low)	THEN (F_{g2} is low)
12.	IF (σ_{g2} is high)	AND (t_g is low)	THEN (F_{g2} is medium)
13.	IF (σ_{g2} is low)	AND (t_g is medium)	THEN (F_{g2} is medium)
14.	IF (σ_{g2} is medium)	AND (t_g is medium)	THEN (F_{g2} is medium)
15.	IF (σ_{g2} is high)	AND (t_g is medium)	THEN (F_{g2} is high)
16.	IF (σ_{g2} is low)	AND (t_g is high)	THEN (F_{g2} is medium)
17.	IF (σ_{g2} is medium)	AND (t_g is high)	THEN (F_{g2} is medium)
18	IF (σ_{g2} is high)	AND (t_g is high)	THEN (F_{g2} is high)

For the C_r parameter:

1.	IF (σ_{g1} is low)	AND (t_g is low)	THEN (C_{rg1} is low)
2.	IF (σ_{g1} is medium)	AND (t_g is low)	THEN (C_{rg1} is low)
3.	IF (σ_{g1} is high)	AND (t_g is low)	THEN (C_{rg1} is medium)
4.	IF (σ_{g1} is low)	AND (t_g is medium)	THEN (C_{rg1} is medium)
5.	IF (σ_{g1} is medium)	AND (t_g is medium)	THEN (C_{rg1} is medium)
6.	IF (σ_{g1} is high)	AND (t_g is medium)	THEN (C_{rg1} is high)
7.	IF (σ_{g1} is low)	AND (t_g is high)	THEN (C_{rg1} is medium)
8.	IF (σ_{g1} is medium)	AND (t_g is high)	THEN (C_{rg1} is medium)
9.	IF (σ_{g1} is high)	AND (t_g is high)	THEN (C_{rg1} is high)
10.	IF (σ_{g2} is low)	AND (t_g is low)	THEN (C_{rg2} is low)
11.	IF (σ_{g2} is medium)	AND (t_g is low)	THEN (C_{rg2} is low)
12.	IF (σ_{g2} is high)	AND (t_g is low)	THEN (C_{rg2} is medium)
13.	IF (σ_{g2} is low)	AND (t_g is medium)	THEN (C_{rg2} is medium)
14.	IF (σ_{g2} is medium)	AND (t_g is medium)	THEN (C_{rg2} is medium)
15.	IF (σ_{g2} is high)	AND (t_g is medium)	THEN (C_{rg2} is high)
16.	IF (σ_{g2} is low)	AND (t_g is high)	THEN (C_{rg2} is medium)
17.	IF (σ_{g2} is medium)	AND (t_g is high)	THEN (C_{rg2} is medium)
18	IF (σ_{g2} is high)	AND (t_g is high)	THEN (C_{rg2} is high)

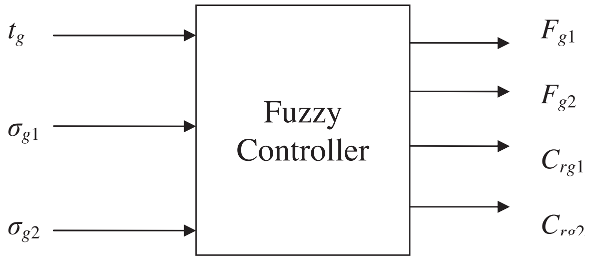


Fig. 8. Inputs and outputs of the fuzzy controller.

With the null hypothesis that there should not have any significant difference between the two WM-DE engines as they are solving the same optimisation problem, the *t*-value should tend to be small along the searching. The rules for the *F* parameter are governed by two inputs: the population standard deviation of the fitness values and the *t*-value. The two inputs are encoded with three linguistic terms. As a result, nine rules can be formulated. Since two WM-DE engines in the intelligent optimiser are controlled by the fuzzy controller, and the control operation are based on the corresponding population standard deviation of fitness values of an engine, totally 18 rules for the *F* parameter can be formulated.

Similarly, another 18 rules can be formulated for the C_r parameter. The C_r parameter affects DE on doing the crossover operation, which modifies the vector elements after the DE mutation operation. It can be considered as introducing some random disturbances to the DE operation. Similar to the *F* parameter setting, we would like to have the solution space to be more widely explored in the early stage of the search, and set a smaller searching space in the later stage of the search to make it more likely to reach a fine-tuned global solution. A large value of C_r means the resulting vector has a bigger difference with the original position, which is preferred in the early stage of the searching process. It can enhance the chance for the population to move away from some local optima. In the later stage, we would like to have a small value of C_r when the population concentrates in the global optimal area so that we can do a fine-tuning for the final solution. With the null hypothesis that there should not have any significant difference between the two WM-DE engines as they are solving the same objective function, the *t*-value should tend to be small along the searching.

The block diagram of the internal fuzzy controller is shown in Fig. 8. This fuzzy controller has 3 inputs and 4 outputs, and generates outputs for every iteration step.

2.2.4.4. WM-DE for multi-objective optimisation. In practice, we often have more than one objective to be optimised in a single problem. A multi-objective optimisation problem can be formulated as follows: optimise $\{f_1(x), f_2(x), \ldots, f_m(x)\}$, where $f_i(x)$; $i = 1, 2, \ldots, m$, is a single objective function; *m* is the number of objective functions of the problem.

```
begin
  Initialise the population for WM-DE Engine 1
  Initialise the population for WM-DE Engine 2
  while (not termination condition) do
    begin
      WM-DE Engine 1 operation for one iteration
      Determine the values of $\alpha_{g1}$ and $\sigma_{g1}$
      WM-DE Engine 2 operation for one iteration
      Determine the values of $\alpha_{g2}$: and $\sigma_{g2}$
      Calculate $t_g$ by (22)
      Determinate the new value of $F_{g1}$ by the fuzzy controller
      Determinate the new value of $F_{g2}$ by the fuzzy controller
      Determinate the new value of $C_{rg1}$ by the fuzzy controller
      Determinate the new value of $C_{rg2}$ by the fuzzy controller
    end
end
```

Fig. 9. Pseudo code of the intelligent optimiser.

Optimising several objectives simultaneously without considering the importance of each objective in the problem is called the Pareto-optimisation. Under the Pareto optimisation, if a new solution can improve at least one of the objectives without declining the other objectives, this solution will be considered as a better solution than the original one. If the new solution can improve at least one of the objectives, but decline the other objectives, this solution will be considered as a poor solution and will be discarded. The objective of the Pareto optimisation is to find a set of solutions that is not dominated by any one solution. This idea can be formulated into the selection operation of WM-DE as follows:

$$\mathbf{x}_{i,g+1} = \begin{cases} \bar{\mathbf{u}}_{i,g} & \text{if } \forall i\, f_i(\bar{\mathbf{u}}_{i,g}) \le f_i(\boldsymbol{x}i, g) \\ \mathbf{x}_{i,g} & \text{otherwise.} \end{cases} \tag{28}$$

where $f_i(\cdot)$ is the *i*th objective function. The pseudo code of the resulting multi-objective WM-DE is shown in Fig. 5. It ensures that WM-DE does not bias against any one of the objective functions in the problem.

In this paper, the final output of the multi-objective WM-DE is a set of non-dominated solutions. The detail on selecting the final solution of the optimiser is discussed in the next section.

2.2.4.5. Training and validation. In the proposed system, WM-DE is employed to optimise the fuzzy inference system (FIS) by finding out the best parameters for the fuzzy rules and membership functions. The job of the FIS is to classify the hypoglycaemic episodes accurately. To measure the performance of the biomedical classification, sensitivity and specificity are considered. The sensitivity measures the proportion of actual positives that are correctly identified, and the specificity measures the proportion of actual negatives that are correctly identified. The definitions of the sensitivity (ξ) and the specificity (η) are given as follows.

$$\xi = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{29}$$

$$\eta = \frac{N_{TN}}{N_{TN} + N_{FP}} \tag{30}$$

where N_{TP} is the number of true positives, which implies the sick people are correctly diagnosed as sick; N_{FN} is the number of false negatives, which implies the sick people are wrongly diagnosed as healthy; N_{FP} is the number of false positives which implies the healthy people are wrongly diagnosed as sick; and N_{TN} is number of true negatives which implied healthy people are correctly diagnosed as healthy [14,15]. The values of sensitivity (ξ) and the specificity (η) are within 0–1. If the value of sensitivity is equal to 1, it means that the classification process is able to classify all the sick people correctly. If the value of specificity is equal to 1, it means that the classification process is able to classify all the healthy people correctly. The sensitivity and the specificity evaluate the performance of the system, which are important criteria for tuning the FIS.

To perform the detection of hypoglycaemic episodes accurately, the proposed system should be able to obtain high values of sensitivity and specificity. The FIS learns the characteristic of the relationship between the system inputs and output through some known dataset. Traditionally, the dataset is divided into two sets, one is used for training the system and the other is used for testing the trained system. However, on using this approach, a phenomenon called overtraining may occur easily. Overtraining refers to the reduction of the generalisation ability that can occur as the system is trained. To overcome this problem, a validation strategy is proposed in this paper to reduce the risk of overtraining. Hence, a validation set is introduced such that the dataset is divided into three sets. While the training set is used to train the FIS, the validation set is used to validate the training result. Then, the testing set is used to test the trained system performance. The main objective of introducing the validation set is to balance the training effect on both the training set and the validation set. As a result, it can avoid the training result being obtained from over fitting with a single data set only.

To train the system with the known dataset, a fitness function should be used to guide the training process and achieve good performance. A proper choice of the fitness function can help the training process achieve a good result. As the validation process is introduced, the number of fitness functions needed to be optimised is increased under the multi-objective approach. We consider four fitness functions for training the FIS.

$$f_1 = \begin{cases} \xi_{target} & \text{if} \quad \xi_{train} \geq \xi_{target} \\ \xi_{train} & \text{otherwise} \end{cases} \tag{31}$$

$$f_2 = \begin{cases} \eta_{target} & \text{if} \quad \eta_{train} \geq \eta_{target} \\ \eta_{train} & \text{otherwise} \end{cases} \tag{32}$$

$$f_3 = \begin{cases} \xi_{target} & \text{if} \quad \xi_{val} \geq \xi_{target} \\ \xi_{val} & \text{otherwise} \end{cases} \tag{33}$$

$$f_4 = \begin{cases} \eta_{target} & \text{if} \quad \eta_{val} \geq \eta_{target} \\ \eta_{val} & \text{otherwise} \end{cases} \tag{34}$$

where ξ_{train} is the sensitivity of the training set; ξ_{val} is the sensitivity of the validation set; ξ_{target} is the target value of sensitivity; η_{train} is the specificity of the training set; η_{val} is the specificity of the validation set; η_{target} is the target value of specificity. The maximum value of each fitness function is equal to 1. These fitness functions are used to train both the training set and validation set for approaching the target values as much as possible. In the proposed system, the target values of the sensitivity and specificity (ξ_{target} and η_{target}) are not constant. They are dynamically updated by some conditions along the search process of the proposed intelligent optimiser as given by the pseudo code in Fig. 6. At the beginning of the optimisation, some initial values are assigned to the two target values, which are both set at 0.10 in this paper. If the sensitivity of both the training set (ξ_{train}) and validation set (ξ_{valid}) meet the target value of sensitivity (ξ_{target}), ξ_{target} will be increased by 0.01. The same method is also applied to the target value of specificity (η_{target}). The updating of ξ_{target} and η_{target} continue until the end of the training and validation process. The objective of this operation is to balance the training effect of both the training set and validation set. For instance, if currently $\xi_{target} = 0.65$, $\xi_{train} = 0.68$ and $\xi_{val} = 0.88$; according to the training and validation process both f_1 and f_3 are kept at 0.65 and ξ_{target} for the next iteration is changed to 0.66. We can see that although ξ_{val} is much larger that ξ_{train}, the proposed intelligent optimiser takes it as the same. As a result, the training process will not bias against the training set or the validation set. The overtraining problem can be alleviated. A more reliable FIS could then be developed.

After the training and validation process, the output of the intelligent optimiser is a group of non-dominated solutions. The solutions are evaluated with the training set and the validation set (the combined set). The solution with the smallest difference between the ξ_{target} and the η_{target} is selected as the final solution (the final output of the optimiser). The difference between the ξ_{target} and the η_{target} of the ith solution is determined as follows:

$$Diff_i = |\xi_i - \xi_{target}| + |\eta_i - \eta_{target}|, \quad i = 0, 1, \ldots, 2N_p - 1 \tag{35}$$

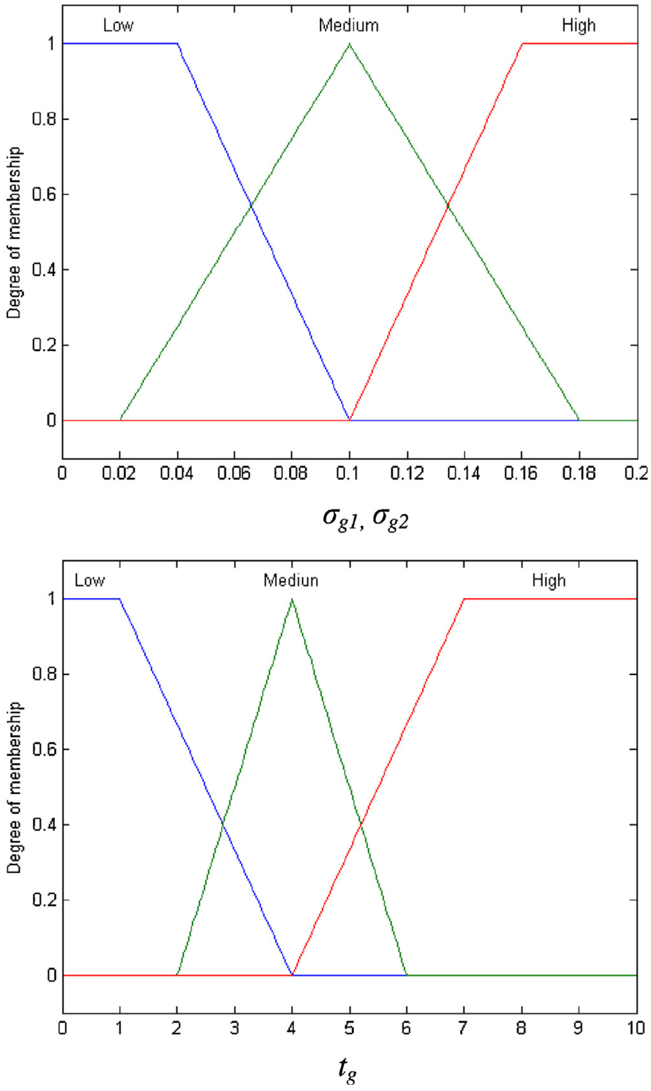


Fig. 10. Input membership functions.

where ξ_i is the sensitivity of the combined set with the ith solution; η_i is the specificity of the combined set with the ith solution.

3. Experiment results and discussions

Sixteen children with TIDM (14.6 ± 1.5 years) volunteered to join a 10-h overnight hypoglycaemia study at the Princess Hospital for Children in Perth, Western Australia, Australia. Each patient was monitored overnight for the natural occurrence of nocturnal hypoglycaemia. Data were collected with approval from Woman's and Children's Health Service, Department of Heath, Government of Western Australia, and with informed consent.

In this study, we measure the required physiological parameters, while the actual blood glucose levels (BGL) are collected as reference using the Yellow Spring Instruments. The signals used for the detection of hypoglycaemia are the HR, QT_c, ΔHR and ΔQT_c. The actual blood glucose profiles for 16 TIDM children are shown in Fig. 12. In this figure, all the data are taken at the same starting time and the duration of data taking for each patient is around 400–480 min. The responses from 16 TIDM children show significant blood glucose changes during the hypoglycaemia phase against the non-hypoglycaemia phase. Normalisation is used to reduce the patient-to-patient variability and to enable group

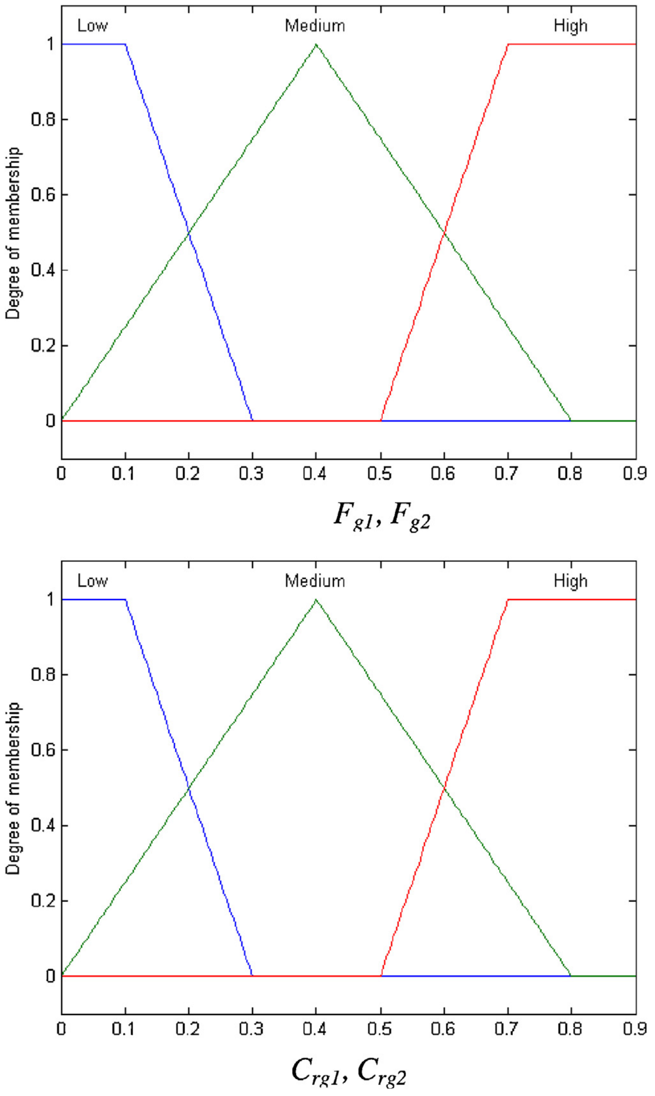


Fig. 11. Output membership functions.

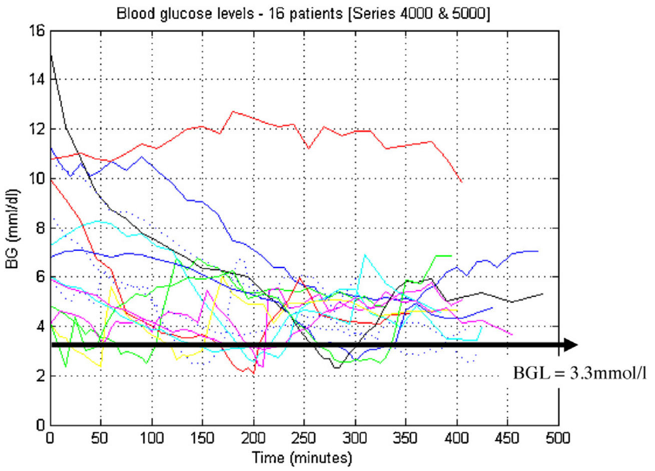


Fig. 12. Actual BG-level profiles in 16 T1DM children.

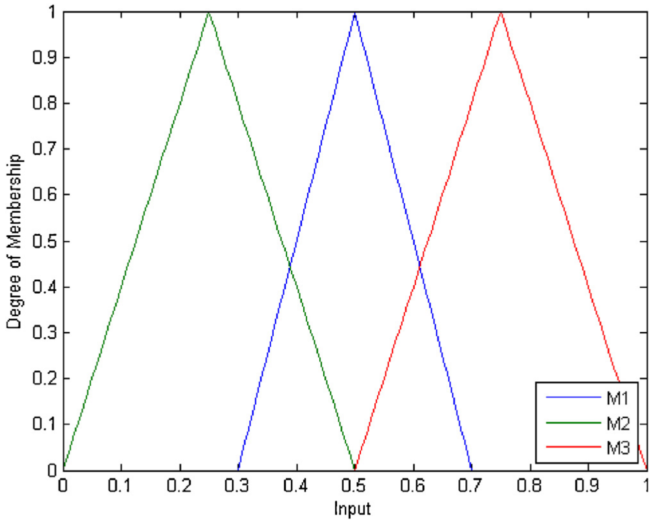


Fig. 13. Triangular-shaped membership functions.

comparison. It is done by dividing the patient's heart rate and corrected *QT* interval by his/her corresponding values at time zero.

In this paper, the meaning of hypoglycaemia is suggested as BGL < 3.3 mmol/l. The detection of hypoglycaemia episodes using these variables is based on a fuzzy inference system trained by the obtained clinical dataset. In effect, it estimates the presence of hypoglycaemia at a sampling period, based on the currently sampled data and the data sampled one period before. The sampling period is 5–10 min and approximately 35–40 data points from each patient were used. The whole data set is divided into a training set, a validation set and a testing set; each with 5 patients randomly selected. With these 15 patients, the number of data points for training, validation, and testing are 199, 177, and 193 respectively. The whole dataset, which included both the hypoglycaemia data part and non-hypoglycaemia data part, is used to measure the classification performance in terms of sensitivity and specificity.

For comparison and analysis purposes, 11 different approaches [10,13,19,22,24] are used to tackle the TIDM problem. They are:

(i) The proposed FIS with four inputs (HR, QT_c, ΔHR and ΔQT_c) tuned by the proposed *intelligent optimiser* with the proposed training and validation method.

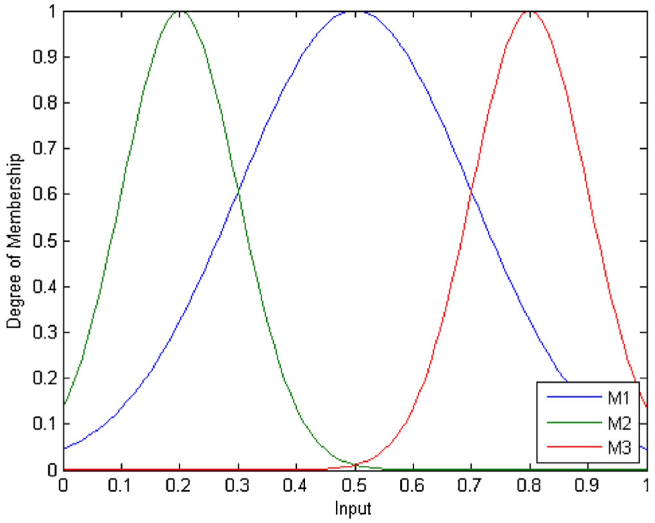


Fig. 14. Bell-shaped fuzzy membership functions.

Table 1a
Results of the proposed system (average of 50 trials).

No of inputs	m_f	t_d	Testing		Training		Validation	
			ξ (%)	η (%)	ξ (%)	η (%)	ξ (%)	η (%)
2	3	21	71.07	40.92	80.35	40.84	83.11	42.14
	5	45	73.22	40.13	80.56	42.21	81.22	41.24
	8	96	73.07	40.21	81.21	43.32	80.00	42.23
4	3	105	72.06	42.22	82.34	41.00	86.24	44.92
	5	665	73.92	51.22	83.33	41.00	90.10	50.95

Table 1b
Results of the WM-DE (average of 50 trials).

No of inputs	m_f	t_d	Testing		Training		Validation	
			ξ (%)	η (%)	ξ (%)	η (%)	ξ (%)	η (%)
2	3	21	70.23	40.12	77.91	40.22	78.11	40.43
	5	45	71.00	40.34	73.02	40.23	76.22	40.31
	8	96	69.12	40.11	78.21	40.19	74.08	40.12
4	3	105	67.54	41.76	76.74	40.90	73.38	40.02
	5	665	68.64	45.72	75.53	40.60	73.05	40.27

Table 2
Best testing results for hypoglycemic detection from different approaches with sensitivity of around 75%.

Method	Sensitivity (%)	Specificity (%)
The *intelligent optimiser* with FIS – 4 inputs	*76.92*	*56.14*
The *intelligent optimiser* with FIS – 2 inputs	75.12	45.32
The *DWM-DE* with FIS – 4 inputs	75.92	55.14
The *DWM-DE* with FIS – 2 inputs	74.92	47.12
NN-RD	79.11	52.01
FIS-4-w/o-v	75.00	51.64
FIS-2-w/o-v	73.21	52.58
LR-4	51.78	51.64
LR-2	50.00	51.17
FFNN-2	64.26	52.50
MR-2	62.31	53.10

The italic values mean the best result within the table.

(ii) An evolved fuzzy inference system with two inputs (*HR* and QT_c) tuned by the proposed *intelligent optimiser* with the proposed training and validation method.
(iii) The proposed FIS with four inputs (*HR*, QT_c, ΔHR and ΔQT_c) tuned by DWM-DE [22].
(iv) An evolved fuzzy inference system with two inputs (*HR* and QT_c) tuned by DWM-DE [22].
(v) A fuzzy inference system with four inputs (*HR*, QT_c, ΔHR and ΔQT_c) without validation (FIS-4-w/o-v) [19].
(vi) A fuzzy inference system with two inputs (*HR* and QT_c) without validation (FIS-2-w/o-v) [19].
(vii) Neural network based rule discovery system with 4 inputs (*HR*, QT_c, ΔHR and ΔQT_c) (NN-RD) [24]
(viii) A linear multiple regression with 4 inputs (*HR*, QT_c, ΔHR and ΔQT_c) (LR-4) [19].
(ix) A linear multiple regression with 2 inputs (*HR* and QT_c) (LR-2) [19].
(x) An evolved multiple regressions with two inputs (EMR2) [19].
(xi) A feed-forward neural network (FFNN) [19].

In (i) and (ii), the proposed intelligent optimiser is used to optimise the fuzzy rules and membership functions of the FIS. The settings of the parameters are given as follows.

- Shape parameter of the wavelet mutation ((ζwm) : 1).
- Parameter λ for the monotonic increasing function: 10,000.
- Initial population: it is generated uniformly at random.
- Maximum number of iteration: 5000
- Population size: 50 × 2

For the two input FIS, three different numbers of membership functions (m_f) have been used for testing. They are three, five, and eight. For the 4-input case, only three and five have been used. In the 2-input case with $m_f = 3$, 5 and 8, the numbers of rules are equal to 9, 25 and 64 respectively. In the 4-input case with $m_f = 3$ and 5, the numbers of rules are equal to 81 and 625 respectively.

The averaged results out of 50 trials of the FIS with the proposed intelligent optimiser under different numbers of inputs and membership functions are shown in Table 1a. The sensitivity and specificity of the training dataset, validation dataset and testing dataset are reported in this table. The averaged results out of 50 trials of the FIS with WM-DE (optimiser without the two populations evolving in parallel and the fuzzy controller) under different numbers of inputs and membership functions are shown in Table 1b. The sensitivity and specificity of the training dataset, validation dataset and testing dataset are reported in this table. By comparing Tables 1a and 1b, we can see that by introducing the population analysis and the fuzzy controller, better performance can be achieved.

The best results out of the 50 trials are given in Table 2. We can see that the FIS model trained with the proposed intelligent optimiser can offer more than 75% in sensitivity and 55% in specificity, which satisfy the common clinical classification requirements (sensitivity > 70% and specificity > 50%). With the number of inputs increased, more information could be offered to the FIS to perform better classification. Moreover, the classification performance of the FIS increases when the number of membership functions increases thanks to the greater degree of freedom of the FIS. Despite the above advantages, the numbers of parameters for tuning will also increase rapidly. It makes the training process to be more computationally demanding. The proposed intelligent optimiser is employed to handle the training process in order to improve the solution quality, solution reliability and the rate of convergence.

4. Conclusion

In this paper, a fuzzy inference system (FIS) is developed to classify hypoglycaemic episodes. The FIS models the relationship between the heart rate, corrected *QT* interval of the electrocardiogram (ECG) signal, their changes, and the hypoglycaemic episodes to perform the classification. A proposed intelligent optimiser embedded with 2 wavelet-mutated differential evolution algorithm is employed to optimise the parameters in the FIS. An experiment using data of 15 children (569 data points) with TIDM is studied. The classification result shows that the proposed FIS performs well and is able to meet the common clinical classification requirement. To conclude, the FIS could possibly model the relationship between the episodes of hypoglycaemia and the physiological signals of heart rate (*HR*), corrected *QT* interval of the electrocardiogram (ECG) signal (QT_c), change of *HR* and change of QT_c. In the future, other modelling methods could be studied for the hypoglycaemic episodes classification problem. For instance, Support Vector Machine (SVM) and Fuzzy Neural Network (FNN) could be used to do the modelling and perform the classification.

Acknowledgements

The work described in this paper was substantially supported by grants from The Hong Kong Polytechnic University (Project No. RP9L) and Juvenile Diabetes Research Foundation International.

References

[1] J.F. Yale, Nocturnal hypoglycemia in patients with insulin-treated diabetes, Diabetes Res. Clin. Pract. 65 (2004) S41–S46.

[2] DCCT Research Group, Adverse events and their association with treatment regimens in the diabetes control and complications trial, Diabetes Care 18 (1995) 1415–1427.

[3] J.C. Pickup, Sensitivity glucose sensing in diabetes, Lancet 355 (2000) 426–427.

[4] Directnet Study Group, Evaluation of factors affecting CGMS calibration, Diabetes Technol. Ther. 8 (2006) 318–325.

[5] F.F.R. Maia, L.R. Araujo, Efficacy of continuous glucose monitoring system (CGMS) to detect postprandial hyperglycemia and unrecognized hypoglycemia in type1 diabetic patients, Diabetes Res. Clin. Pract. 75 (2007) 30–34.

[6] O. Amir, D. Weinstein, S. Ziberman, M. Less, D. Perl-Treves, H. Primack, A. Weinstein, E. Gabis, B. Fikhte, A. Karasik, Continuous noninvasive glucose monitoring technology based on occlusion spectroscopy, J. Diabetes Sci. Technol. 1 (4) (2007) 463–469.

[7] R.L. Weinstein, Accuracy of the freestyle navigator CGMS: comparison with frequent laboratory measurements, Diabetes Care 30 (2007) 1125–1130.

[8] A. Caduff, M.S. Talary, M. Mueller, F. Dewarrant, J. Klisic, M. Donath, L. Heinemann, W.A. Stahel, Non-invasive glucose monitoring in patient with type 1 diabetes: a multisensory system combining sensors for dielectric and optical characterization of skin, Biosens. Bioelectron. 24 (2009) 2778–2784.

[9] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimisation over continuous spaces, J. Global Optim. 11 (1997) 341–359.

[10] G.A.F. Seber, Linear Regression Analysis, Wiley, New York, 2003.

[11] B.H. Cho, H. Yu, K.W. Kim, T.H. Kim, I.Y. Kim, S.I. Kim, Application of irregular and unbalanced data to predict diabetic nephropathy using visualization and feature selection methods, Artif. Intel. Med. 42 (1) (2008) 37–53.

[12] A. Chu, H. Ahn, B. Halwan, B. Kalmin, E.L.V. Artifon, A. Barkun, M.G. Lagoudakis, A. Kumar, A decision support system to facilitate management of patients with acute gastrointestinal bleeding, Artif. Intel. Med. 42 (3) (2008) 247–259.

[13] S. Wang, W. Min, A new detection algorithm (NDA) based on fuzzy cellular neural networks for white blood cell detection, IEEE Trans. Inf. Technol. Biomed. 10 (January (1)) (2006) 5–10.

[14] D.A. Freedman, Statistical Models: Theory and Practice, Cambridge Univ. Press, Cambridge, UK, 2005.

[15] D.G. Altman, J.M. Bland, Statistics notes: diagnostic tests 1: sensitivity and specificity, Br. Med. J. 308 (1994) 1552.

[16] I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[17] U.K. Chakraborty (Ed.), Advances in Differential Evolution, Springer, Heidelberg, 2008.

[18] H.Y. Fan, J. Lampinen, A trigonometric mutation operation to differential evolution, J. Global Optim. 27 (1) (2003) 105–129.

[19] S.H. Ling, H.T. Nguyen, Genetic-algorithm-based multiple regression with fuzzy inference system for detection of nocturnal hypoglycemic episodes, IEEE Trans. Inform. Technol. Biomed. 15 (2) (2011) 308–315.

[20] K.Y. Chan, C.K. Kwong, T.S. Dillon, Y.C. Tsim, Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming, Appl. Soft Comput. 11 (2) (2011) 1648–1656.

[21] K.Y. Chan, T.S. Dillon, C.K. Kwong, Modeling of a liquid epoxy molding process using a particle swarm optimisation-based fuzzy regression approach, IEEE Trans. Ind. Inform. 7 (1) (2011) 148–158.

[22] J.C.Y. Lai, F.H.F. Leung, S.H. Ling, H.T. Nguyen, Hypoglycaemia detection using fuzzy inference system with multi-objective double wavelet mutation differential evolution, Appl. Soft Comput. 13 (5) (2013) 2803–2811.

[23] F.H.F. Leung, S.H. Ling, E. Shi, An improved differential evolution and its industrial application, J. Intel. Learn. Syst. Appl. 4 (2) (2012) 81–97.

[24] K.T. Chan, S.H. Ling, T.S. Dillon, H.T. Nguyen, Diagnosis of hypoglycemic episodes using a neural network based rule discovery system, Expert Syst. Appl. 38 (8) (2011) 9799–9808.

[25] J.C.Y. Lai, F.H.F. Leung, S.H. Ling, E. Shi, An improved differential evolution and its industrial application, J. Intel. Learn. Syst. Appl. 4 (2) (May 2012) 81–97.