Thalassemic Patient Classification Using a Neural Network and Genetic Programming

Waranyu Wongseree* and Nachol Chaiyaratana**

บทคัดย่อ

บทความนี้แสดงการใช้ระบบการโปรแกรมเชิงพันธุกรรมที่เรียกว่า STROGANOFF และมัลติเลเยอร์เพอร์เซ็บตรอน ในการจำแนกประเภทของผู้ป่วยโรคธาลัสซีเมีย ปัญหาที่สนใจครอบคลุมตัวอย่างทดสอบจากคนปกติ ผู้ป่วยและพาหะโรคธาลัสซีเมีย ข้อมูลเข้าที่ใช้ในการจำแนก ได้แก่ คุณลักษณะเฉพาะของเม็ดเลือดแดง เม็ดเลือดแดงตัวอ่อน และเกล็ดเลือด ที่ได้จากตัวอย่างทดสอบ ผลของการทดลองแสดงให้เห็นว่าประสิทธิภาพของต้นไม้จำแนกที่สร้างโดยการโปรแกรมเชิงพันธุกรรมมีความใกล้เคียงกับประสิทธิภาพของมัลติเลเยอร์เพอร์เซ็บตรอนที่มีชั้นซ่อนหนึ่งชั้น ในทางตรงกันข้ามมัลติเลเยอร์เพอร์เซ็บตรอนที่มีชั้นซ่อนสองชั้นมีประสิทธิภาพสูงกว่าต้นไม้จำแนกที่สร้างโดยการโปรแกรมเชิงพันธุกรรม อย่างไรก็ตามโครงสร้างของต้นไม้จำแนก แสดงให้เห็นว่าคุณลักษณะเฉพาะของเกล็ดเลือดไม่มีผลต่อประสิทธิภาพในการจำแนก ด้วยเหตุนี้จึงเป็นไปได้ที่จะลดจำนวนข้อมูลเข้าสำหรับการจำแนกและปรับปรุงประสิทธิภาพในการจำแนกต่อไป

Abstract

This paper presents the use of a genetic programming (GP) system called STROGANOFF and a multilayer perceptron for thalassemic patient classification. The interested problem covers the test samples from normal subjects and that from different types of thalassemic patient and thalassemic trait. The features, which are the characteristics of red blood cell, reticulocyte and blood platelet extracted from the blood samples, are used as input to the classifiers. The results indicate that the performance of the GP-generated classification trees is approximately equal to that of the multilayer perceptrons with one hidden layer. In contrast, the multilayer perceptrons with two hidden layers outperform GP-generated classification trees. Nonetheless, the structure of the classification trees reveals that the characteristics of blood platelet have no effects on the classification performance. This helps to reduce the required input features for the task and make further improvements possible.

Keywords: Genetic Programming, Neural Network, Pattern Recognition, Thalassemia.

1. Introduction

Thalassemia is a disease that causes a reduction in the life span of a red blood cell [1]. The disease is a result of an abnormality in the genes that regulate the formation of haemoglobin, which is a core component of the red blood cell. Hence, this disease is also hereditary. The disease is very common in tropical and subtropical areas. For instance, in Thailand at least 1% of the population has the disease and approximately 30-40% of the population is thalassemic trait. In order to make the

* Department of Electrical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology North Bangkok.

** Research and Development Center for Intelligent Systems, King Mongkut's Institute of Technology North Bangkok.

diagnosis, the blood characteristics must be analysed. However, with a large number of possible candidate characteristics together with variety in types of the disease and trait, a manual diagnostic process can only be carried out by specialists. Although an automated diagnostic tool has been previously developed for the task [2-3], the available rule-based tool covers a much broader range of blood-related diseases including various types of anaemia. In order to narrow the diagnostic target down to the differentiation between thalassemic patients, thalassemic traits and normal subjects, an alternative automated diagnostic tool is required.

The thalassemic patient classification problem can generally be formulated into a pattern recognition problem. The input patterns or samples in this case would be blood-related features that are the characteristics of red blood cell, reticulocyte and blood platelet extracted from the blood samples. On the other hand, the target classification output would be either the disease/trait type or a flag indicating that the subject is normal. The use of a neural network and a set of genetic programming (GP) based decision trees as classifiers is proposed. In the case of the neural network, a multilayer perceptron with a back-propagation algorithm will be used where a suitable network configuration is obtained by varying the number of hidden nodes and hidden layers. In contrast, the structure of the GP-based decision tree is determined solely by means of simulated evolution. Specifically, the GP system chosen for implementing the decision tree is a technique called a structured representation on genetic algorithms for non-linear function fitting or STROGANOFF [4]. The technique dictates the use of a binary tree where each node in the tree is defined by a second order polynomial function. Furthermore, the use of a multiple regression analysis method as a means for identification of the polynomial coefficients is also recommended. Note that this regression analysis method is generally referred to as a group method for data handling (GMDH) where it is based on a statistical method used in system identification [5]. In order to regulate the size and structure of the decision tree, a minimum description length (MDL) criterion is embedded into the GP-based search strategy. With the use of the MDL criterion, the decision tree will be constructed by considering a trade-off between the classification accuracy and the tree size. This helps to promote the survival of a decision tree that has a reasonably small size while also maintains high classification accuracy.

This paper is organised as follows. The interested thalassemic patient classification problem is explained in section 2. In section 3, the background on the STROGANOFF and the minimum description length criterion is described. Finally, the results obtained from using the GP-based decision tree and multilayer perceptron as the thalassemic patient classifiers, the related discussions and the conclusions are given in section 4.

2. Thalassemic Patient Classification Problem

Thalassemia, which is a form of anaemia, is a disease that causes a reduction in the working life of a red blood cell [1]. The disease is a result of an abnormality on the genes that regulate the formation of a protein called globin, which is a major component of haemoglobin (Hb). Note that haemoglobin is in turns a core component of a red blood cell where each red blood cell contains approximately 300 million molecules of haemoglobin. Hence, a change in the structure of globin would affect the structure and functionality of a red blood cell. A globin molecule contains two parts: α-globin and β-globin. The α-globin contains 141 amino acids, which are regulated by genes on chromosome 16. On the other hand, the β-globin consists of 146 amino acids, which are governed by a gene on chromosome 11.

The abnormality in the construction of globin, which leads to the formation of abnormal haemoglobin, can be categorized into two types: an abnormality

in the types of amino acids that are parts of haemoglobin and an abnormality that leads to a reduction of the globin production. Various types of abnormal haemoglobin that falls into the first category have been detected among population in tropical and sub-tropical areas. However, the types that frequently occur in Thai population are haemoglobin E (Hb E) and haemoglobin constant spring (Hb CS). The haemoglobin E is haemoglobin with an abnormality in β-globin while the haemoglobin constant spring is haemoglobin with an abnormality in α-globin. In the case of the second category where the abnormality can lead to thalassemia, the two main types of abnormality are α-thalassemia and β-thalassemia. As the name stated, the α-thalassemia is caused by an abnormality of α-globin. The α-thalassemia can be further divided into two types: α-thalassemia 1 or α^0-thalassemia and α-thalassemia 2 or α^+-thalassemia. The α-thalassemia 1 refers to a complete absent of α-globin while the α-thalassemia 2 refers to a reduction in the α-globin production. Similarly, the β-thalassemia is caused by an abnormality of β-globin and can also be further divided into two types: β^0-thalassemia and β^+-thalassemia. As the notation implied, the β^0-thalassemia refers to a complete absent of β-globin while the β^+-thalassemia refers to a reduction in the β-globin production. Note that a haemoglobin E abnormality can also lead to β-thalassemia since the genetic defect that causes the change of amino acids in the β-globin structure also reduces the β-globin production. On the other hand, a haemoglobin constant spring abnormality can also lead to α-thalassemia since the genetic defect also decreases the production of α-globin.

Since the genes that regulate the construction of globin reside on two autosomes (chromosomes 11 and 16), the genes that cause an abnormality in haemoglobin are autosomal. In addition, it has been proven that the thalassemic genes are also recessive genes. This means that in order for a person to have thalassemia, that person must have two copies of a gene on the same chromosome that are recessive. Specifically, one of the gene copies must directly cause either α-thalassemia 1 or β-thalassemia. For instance, a patient who suffers from a haemoglobin H disease would have two copies of a thalassemic gene on chromosome 16 where one copy of the gene causes an α-thalassemia 1 abnormality while the other copy causes either α-thalassemia 2 or haemoglobin constant spring abnormalities. Note that these abnormalities occur on the α-globin molecule. On the other hand, a person with one copy of a gene that causes α-thalassemia and one copy of another gene that causes β-thalassemia would not have the disease since the genes regulate different parts of the globin structure. In addition, the expression of both thalassemic genes will also be suppressed since the other copies of the corresponding genes are normal and dominant. Similarly a person with two copies of a gene that causes either a haemoglobin E abnormality (homozygous Hb E) or α-thalassemia 2 (homozygous α-thalassemia 2) would not exhibit any symptoms of thalassemia since neither copy of the genes directly causes α-thalassemia 1or β-thalassemia.

The key factors that can be used to identify the disease are the type of recessive gene and the number of recessive gene copies that a person carries. In the case that a person has only one copy of recessive gene, that person is generally referred to as a trait. Although a trait would never have the disease, some traits would present a direct risk to their offspring. For instance if both parents are α-thalassemia 1 traits, there is a 25% chance that their offspring would have the most serious form of the disease called Hb Bart's hydrops fetalis. This form of thalassemia leads to either a stillbirth or an infancy death since the offspring completely fails to produce α-globin, which implies that the offspring is incapable of constructing a red blood cell. It is obvious that the most reliable method for the diagnosis is to determine the genetic makeup of the person in question. However, this process is costly and time-

Table 1 Candidate features for the diagnosis

Feature	Description
	Red blood cell (mature)
1	Red blood cell count
2	Haemoglobin concentration
3	Haematocrit
4	Mean corpuscular volume
5	Mean corpuscular haemoglobin
6	Mean corpuscular haemoglobin concentration
7	Cellular haemoglobin concentration mean
8	Red blood cell distribution width
9	Haemoglobin distribution width
	Platelet
10	Platelet count
11	Platelet count after a calculation correction
12	Mean platelet volume
13	Platelet distribution width
14	Plateletcrit
	Reticulocyte
15	Reticulocyte count
16	Reticulocyte percentage
17	Mean corpuscular volume
18	Cellular haemoglobin concentration mean
19	Red blood cell distribution width (reticulocyte)
20	Haemoglobin distribution width

Table 2 Sample groups used to train and test the classifiers

Group	Description	Targeting Globin	# Gene Copies	# Samples
1	α-thalassemia 1 trait	α-globin	1	26
2	α-thalassemia 2 trait	α-globin	1	24
3	β-thalassemia trait	β-globin	1	21
4	Hb constant spring trait	α-globin	1	12
5	Hb E trait	β-globin	1	53
6	Homozygous Hb E	β-globin	2	11
7	Hb H disease	α-globin	2	28
8	Hb H disease with constant spring	α-globin	2	24
9	Homozygous β-thalassemia	β-globin	2	12
10	Normal subject	-	-	59

consuming. An alternative approach is to make a diagnosis based upon the characteristics of human blood. Three major components of blood that can be used in this case are 1) a mature red blood cell that in general is simply referred to as a red blood cell, 2) a young red blood cell or a reticulocyte and 3) a blood platelet. The charac-teristics or features that can be extracted from these three blood components are summarised in Table 1.

With a wide range of candidate characteristics being available, the use of an automated diagnostic tool is required. In this paper, a multilayer perceptron and a genetic programming (GP) based decision

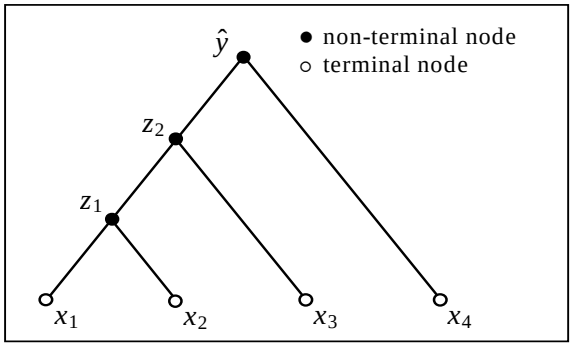

Figure 1 A binary tree with four terminals.

tree will be used as classifiers. Specifically, the GP system chosen for this paper is a technique called a structured representation on genetic algorithms for non-linear function fitting or STROGANOFF [4]. Detailed explanation of the STROGANOFF will be given in the next section.The multilayer perceptron and the GP-based decision tree will be used to classify 10 groups of samples: 9 groups of samples from persons with recessive genes and 1 group of samples from normal subjects. The description of the sample groups is summarised in Table 2. These samples will be used as training and testing data where the classifier set up will be explained in section 4.

3. STROGANOFF and MDL Criterion

One of the classifiers, which will be used in this paper, is a GP-based decision tree. The GP system chosen for the implementation of the decision tree is called a structure representation on genetic algorithms for non-linear function fitting or STROGANOFF [4]. The STROGANOFF is capable of being used as a universal approximator in a similar way to a neural network. However, the technique dictates the use of a binary tree where each non-terminal node is modelled by a second order polynomial. Without loss of generality, assuming that the STROGANOFF is used to approximate a function where an input-output data set is available. Consider a binary tree given in Figure 1. In Figure 1, the binary tree takes four terminals or inputs $\{x_1, x_2, x_3, x_4\}$ to produce an approximate output $\hat{y}$.

Table 3 Bivariate basis polynomials

i	Polynomial Function
1	$f_1(x_1,x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$
2	$f_2(x_1,x_2) = a_0 + a_1x_1 + a_2x_2$
3	$f_3(x_1,x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2$
4	$f_4(x_1,x_2) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_1^2$
5	$f_5(x_1,x_2) = a_0 + a_1x_1 + a_2x_2^2$
6	$f_6(x_1,x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2$
7	$f_7(x_1,x_2) = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2^2$
8	$f_8(x_1,x_2) = a_0 + a_1x_1^2 + a_2x_2^2$
9	$f_9(x_1,x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2$
10	$f_{10}(x_1,x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2$
11	$f_{11}(x_1,x_2) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_1^2 + a_4x_2^2$
12	$f_{12}(x_1,x_2) = a_0 + a_1x_1x_2 + a_2x_1^2 + a_3x_2^2$
13	$f_{13}(x_1,x_2) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_2^2$
14	$f_{14}(x_1,x_2) = a_0 + a_1x_1 + a_2x_1x_2$
15	$f_{15}(x_1,x_2) = a_0 + a_1x_1x_2$
16	$f_{16}(x_1,x_2) = a_0 + a_1x_1x_2 + a_2x_1^2$

This tree can be written as a (LISP) S-expression, (NODE1(NODE2(NODE3(x_1)(x_2))(x_3))(x_4)) where NODE3 and NODE2 represent intermediate variables z_1 and z_2, respectively and NODE1 denotes the output $\hat{y}$. The intermediate variables z_1 and z_2 and the output $\hat{y}$ are given by

$$z_1 = f_i(x_1, x_2), \tag{1}$$

$$z_2 = f_i(x_3, z_1) \tag{2}$$

and $$\hat{y} = f_i(x_4, z_2) \tag{3}$$

where each f_i expression takes one of the sixteen quadratic forms given in Table 3. With the use of a polynomial function in each non-terminal node, coefficients in the polynomial can then be solved using a hierarchical multiple regression analysis method. Referring to the binary tree given in Figure 1, the coefficients in the polynomial $f_i(x_1, x_2)$ are identified by trying to fit z_1 with y - the desired value of output. Subsequently, the coefficients in the polynomial $f_i(x_3, z_1)$ are then identified by fitting z_2 to y. Finally, the coefficients in the polynomial $f_i(x_4, z_2)$ are identified in a similar fashion. Note that this hierarchical multiple regression analysis method is generally referred to as a group method for data handling (GMDH) where it is based on a statistical method used in system identification [5].

Similar to other implementations of genetic programming, the binary tree created by the STROGANOFF is constructed by means of simulated evolution. However, the size of the binary tree in this case is regulated by a minimum description length (MDL) criterion. Instead of using only an error between the desired and actual outputs from the binary tree as the fitness index, the MDL-based fitness function takes both the error and the tree structure into account. In other words, a solution with an acceptable level of modelling/classification error that has a reasonably small tree size would have a high chance of survival. Three MDL-based fitness functions are used in this paper. The first function is taken from Iba et al. [4] where the function can be described as

$$MDL_1 = 0.5A\log N + 0.5N\log(MSE) \tag{4}$$

where A is the number of polynomial coefficients in the tree, N is the total number of input-output samples and MSE is the mean squared error between the target and actual outputs from the tree, which is denoted by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 \tag{5}$$

where y_i is the ith sample of target output and $f(\mathbf{x}_i)$ is the output from the decision tree with this sample $\mathbf{x}_i$. The second MDL-based fitness function is taken from Nikolaev and Iba [6] where the function is given by

$$MDL_2 = MSE + \frac{A}{N}s^2\log(N) \tag{6}$$

where s^2 is a rough estimate of the unknown error variance which is denoted by

$$s^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2 \quad (7)$$

where $\bar{y} = (1/N)\sum_{i=1}^{N} y_i$ is the mean of desired output sample. The last MDL-based fitness function is also taken from Nikolaev and Iba [6] where the function can be described as

$$MDL_3 = RAE + \frac{A}{N} s^2 \log(N) \quad (8)$$

where *RAE* is the regularised average error, which is given by

$$RAE = \frac{1}{N}\left(\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + k\left|\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2}\right|^2\right) \quad (9)$$

where k is the regularised parameter. Note that all three MDL-based fitness functions describe different trade-offs between the error and the tree size. In addition, an MDL value can be calculated for each sub-tree in the complete binary tree.

The MDL criterion can also be used as guidance during the crossover process. Similar to other GA-based systems, an individual obtained after a crossover operation may be affected by a semantic disruption induced by the crossover operator. Iba et al. [7] have suggested that during crossover, a sub-tree with a low MDL value should be replaced by another sub-tree with a higher MDL value from the individual chosen for mating. Along the same line of reasoning, a sub-tree with a low MDL value should also have a higher chance for mutation. Note that a mutation can lead to 1) a change from one terminal (x_i) to another terminal, 2) a change from a terminal to a non-terminal node (sub-tree), 3) a change from a non-terminal node to a terminal and 4) a change of polynomial form used in a non-terminal node. The STROGANOFF has been successfully used in various applications including pattern recognition [4], data mining [8] and time-series prediction [6]. Detailed explanation about the STROGANOFF and the MDL criterion can be found in Nikolaev and Iba [6].

4. Results, Discussions and Conclusions

The thalassemic patient/trait and normal subject data presented in section 2 are used to train and test the neural networks and GP-based decision trees. Firstly, the data are normalised such that the range of feature values is between 0 and 1. Then the centroid of samples in each group is determined. The normalised data set is subsequently divided into one training set and one testing set. The data separation is carried out in the manner that the distribution of Euclidean distances between the training samples and the associated group centroids is similar to that between the testing samples and the centroids. The resulting training set contains 137 samples while the testing set consists of 133 samples.

The neural network that is selected for this implementation is a multilayer perceptron. The multilayer perceptron will be trained using a back-propagation algorithm. Since there are 20 input features and 10 output classes, the number of input and output nodes are set to 20 and 10, respectively. Note that the target network output is in the form of a ten-dimensional unit vector where each vector in the direction of a principal axis of the output vector space represents a target output class. In this paper, the network can have either one hidden layer or two hidden layers. This is done in order to investigate the effect of network non-linearity on the classification performance. The trial numbers of

Table 4 Parameter settings for the multilayer perceptron

Parameter	Value
Number of input nodes	20
Number of output nodes	10
Number of hidden layers	1 or 2
Number of neurons in each hidden layer	5, 10 or 15
Learning rate parameter	0.1
Momentum constant	0.9
Number of training epochs	10,000
Number of repeated runs	10

Table 5 Parameter settings for the STROGANOFF

Parameter	Value
Maximum allowable tree depth	5
Minimum allowable tree depth	2
Maximum possible number of nodes	64
MDL-based fitness function	MDL_1, MDL_2 or MDL_3
Regularised parameter (MDL_3)	0.001
Selection technique	Stochastic universal sampling
Number of elitist individuals	2
Crossover probability	0.6
Mutation probability	0.3
Size of the function set	16 (refer to Table 3)
Size of the terminal set	21
Population size	100
Number of generations	100
Number of repeated runs	10

hidden nodes and other network parameter settings are summarised in Table 4.

As mentioned earlier, the STROGANOFF will be used to generate the decision tree. However, in order to construct a GP-based classifier, which is comparable to the multilayer perceptron, ten separate decision trees are required. Each decision tree will have a role of separating samples that belong to one class from samples that are members of the remaining nine classes. By setting the target output of each tree to either 0 or 1, the resulting decision vectors would have the same format as the output from the multilayer perceptron. The parameter settings for the STROGANOFF are summarised in Table 5. Notice that the number of terminals or input features in Table 5 is set to 21. The additional terminal is the sum of all 20 normalised input features. This is done according to the recommendation given in Nikolaev et al. [9] for the purpose of function over-fitting avoidance during the decision tree construction. Also note that although each tree construction is repeated 10 times, the GP-based classifier will be made up from only 10 decision trees where each tree is the best tree among 10 trees, which are trained to perform the same task.

The classification performance of the multi-layer perceptron (MLP) and GP-based decision tree is summarised in Table 6. From Table 6, it can be seen that the GP-based decision tree that has the

Table 6 Classification accuracy of the multilayer perceptron and GP-based decision tree

Classifier	Training (%)			Testing (%)		
	Max	Mean	S.D.	Max	Mean	S.D.
MLP (in-hid-out)						
20-5-10	94.16	91.24	1.95	80.45	77.44	2.00
20-10-10	97.81	95.62	1.09	84.96	82.86	1.54
20-15-10	97.08	95.47	1.13	85.71	83.08	1.92
20-5-5-10	98.54	95.40	1.58	88.72	83.76	3.77
20-10-10-10	99.27	97.74	1.26	88.72	84.44	2.41
20-15-15-10	99.27	98.61	0.64	86.47	84.74	1.33
GP-based tree						
MDL_1	92.70	-	-	85.71	-	-
MDL_2	83.21	-	-	78.95	-	-
MDL_3	81.02	-	-	75.94	-	-

classification performance in the comparable level to that of the multilayer perceptron with one hidden layer is the one that utilised the MDL-based fitness function given in equation (4). The results also indicate that the GP-based decision trees that use other MDL-based fitness functions have much lower classification accuracy than that of the multilayer perceptron. Nonetheless, after comparing the overall performance of the GP-based decision tree with that of the multilayer perceptron with two hidden layers it is obvious that the most suitable classifier for the investigated problem is the multilayer perceptron. These results can be interpreted as follows.

By inspecting the results among three modes of genetic programming implementation, it is noticeable that the decision trees generated by the STROGANOFF with the MDL_2 and MDL_3 functions have the same level of classification performance. This level of performance is also significantly lower than that of the decision tree, which employs the use of MDL_1 function. As mentioned earlier in section 3, all three MDL-based fitness functions produce different trade-offs between the classification error and the tree size. After inspecting the MDL-based fitness functions, it can be seen that a major difference between the MDL_1 function and the MDL_2/MDL_3 functions lies in the error term. Specifically, the difference is that the mean squared error term in the MDL_1 function is expressed in a logarithmic scale while the same terms in the MDL_2 and MDL_3

functions are given in a normal scale. This difference is most likely to be the main cause of the variation in the classification performance.

Moving onto the performance comparison between the multilayer perceptron and the GP-based decision tree. Firstly, focus on the results generated by the multilayer perceptron with one hidden layer and STROGANOFF with the MDL_1 function. In terms of the classification accuracy during training, the performance of the decision tree is lower than that of the multilayer perceptron. On the other hand, the classification performance of the decision tree and the multilayer perceptron during testing is pretty much the same. These results agree with the classification performance evaluation of the STROGANOFF carried out in Iba et al. [4]. In other words, this is yet another application, which proves that the STROGANOFF can produce a GP-tree that is capable of capturing an input/output relationship during generalisation as well as a multilayer perceptron with one hidden layer. However, once the complexity of multilayer perceptron is increased after the number of hidden layers has changed to two, the neural network clearly outperforms the GP-based decision tree. This is not surprising since it can be said at this point that the equivalent non-linearity level of the GP-tree generated by the STROGANOFF is similar to that of a multilayer perceptron with one hidden layer. Further modification of the STROGANOFF is required in order to increase the non-linearity level of the GP-tree and the associated modelling capability.

Although it is obvious that the best classifier for the interested problem is the multilayer perceptron with two hidden layers, the GP-based decision tree can be used to identify the significance of each feature on the ability to solve the problem. By inspecting the decision tree generated by the STROGANOFF, it is noticeable that the input features or terminals that have disappeared from the evolved decision tree are features 9-15 from Table 1. Features 10-14 make up the full feature set of blood platelet characteristics. On the other hand, feature 9 corresponds to the haemoglobin distribution width for a red blood cell while feature 15 denotes the reticulocyte count. The absent of these features from the decision tree, especially the one generated by the STROGANOFF with the MDL_1 function, signifies that these features have no effects on the diagnosis capability of the GP-based classifier. Since the performance of the decision tree generated with the use of the MDL1 function is very close to that of the multilayer perceptron with one hidden layer, it may also be possible to further improve the classification performance of the multilayer perceptron by removing these features from the network input. Further investigation is required for proving this idea.

5. Acknowledgements

The authors acknowledge the collaboration with Prof. Suthat Fucharoen, M.D. and the research staffs at Thalassemia Research Center, Institute of Science and Technology for Research and Development, Mahidol University, Nakhonpathom, Thailand, in providing the blood sample data.

References

1. Weatherall , D. J. and Clegg, J. B. *The thalassemia syndromes*. (4th ed.). Malden, MA : Blackwell Science, 2001.
2. Lanzola, G. , et al. “NEOANEMIA: A knowledge-based system emulating diagnostic reasoning.” *Computers and Biomedical Research.* 23 (1990) : 560 - 582.
3. Quaglini, S., et al. “ANEMIA : An expert consultation system.” *Computers and Biomedical Research.* 19, 1 (1986) : 13 - 27.
4. Iba, H., H. de Garis and Sato, T. “Genetic programming using a minimum description length principle.” In. K. E. Kinnear, Jr. (Ed.), *Advances in genetic programming*. Cambridge, MA : MIT Press, 1994.
5. Ivakhnenko, A.G. “Polynomial theory of complex

systems." *IEEE Transactions on Systems, Man, and Cybernetics.* 1, 4 (1971) : 364 - 378.

6. Nikolaev, N. and Iba, H. "Regularization approach to inductive genetic programming." *IEEE Transactions on Evolutionary Computation.* 5, 4 (2001) : 359 - 375.
7. Iba, H., Sato , T. and H. de Garis. "Recombination guidance for numerical genetic programming." *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation.* 97-102. Perth, Australia, November 1995.
8. Nikolaev, N. and Iba, H. "Inductive genetic programming of polynomial learning networks." *Proceedings of the 2000 IEEE International Symposium on Combinations of Evolutionary Computation and Neural Networks,* 158-167. San Antonio, TX, May 2000.
9. Nikolaev, N., L. M. de Menezes and Iba, H. "Overfitting avoidance in genetic programming of polynomials." *Proceedings of the 2002 Congress on Evolutionary Computation, 1209-1214.* Honolulu, HI, May 2002.