International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **4285–4297**

SELECTING A SMALL SUBSET OF INFORMATIVE GENES FROM GENE EXPRESSION DATA BY USING A MODIFIED BINARY PARTICLE SWARM OPTIMISATION

Mohd Saberi Mohamad[1,*], Sigeru Omatu[2], Safaai Deris[1]
Michifumi Yoshioka[3] and Zuwairie Ibrahim[4]

[1]Artificial Intelligence and Bioinformatics Research Group
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
81310 Skudai, Johor, Malaysia
[*]Corresponding author: saberi@utm.my

[2]Department of Electronics, Information and Communication Engineering
Osaka Institute of Technology
5-16-1 Omiya, Asahi-ku, Osaka 535-8585, Japan

[3]Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531, Japan

[4]Faculty of Electrical and Electronic Engineering
Universiti Malaysia Pahang
26600 Pekan, Pahang, Malaysia

Received February 2011; revised September 2011

Abstract. *Gene expression technology, especially microarrays, can be used to measure the expression levels of thousands of genes simultaneously in biological organisms. Gene expression data produced by microarrays are expected to be useful for cancer classification. To select a small subset of informative genes for cancer classification, many researchers have analysed the gene expression data using various computational intelligence methods. However, due to the small number of samples compared with the huge number of genes (high-dimensional data), irrelevant genes, and noisy genes, many of the computational methods face difficulties in selecting the small subset. Thus, we propose a modified binary particle swarm optimisation to select a small subset of informative genes that are relevant for the cancer classification. In the proposed method, we introduce the particle speed and a rule for increasing the probability of bits in a particle's position to be zero. The method was empirically applied to a suite of four well-known benchmark gene expression data sets. The experimental results demonstrate that the proposed method outperforms the conventional version of binary particle swarm optimisation (BPSO) and other related works in terms of classification accuracy and the number of selected genes. In addition, this method also produces lower running times compared to BPSO.*
Keywords: Binary particle swarm optimisation, Gene selection, Gene expression data, Cancer classification

1. **Introduction.** Advances in the area of microarray-based gene expression analysis have led to a promising future for the diagnosis using new molecular-based approaches [1]. Microarray technology allows scientists to measure the expression levels of thousands of genes simultaneously, producing gene expression data that contain useful genomic, diagnostic, and prognostic information for researchers [2]. Comparisons between the gene expression levels of cancerous and normal tissues can be performed, and these comparisons

are useful for selecting genes that might predict the clinical behaviour of cancers. Thus, there is a need to select informative genes that contribute to cancerous states. However, the gene selection process poses a major challenge because of the following characteristics of gene expression data: the huge number of genes compared with the small number of samples (high-dimensional data), irrelevant genes, and noisy data. To overcome these challenges, a gene selection method is typically used to select a subset of genes that maximises the ability to classify samples more accurately [3]. Gene selection is called feature selection in the pattern recognition domain, and it has several advantages [4]:

1) It can maintain or improve classification accuracy.
2) It can reduce the dimensionality of the data.
3) It can yield a small subset of genes.
4) It can remove irrelevant and noisy genes.
5) It can reduce the cost in a clinical setting.

In the context of cancer classification, gene selection methods can be grouped into two categories [5]. If a gene selection method is carried out independently from a classification procedure, it belongs to the filter method. Otherwise, it is said to follow a hybrid (wrapper) method. In the early era of gene expression analysis, most research groups used the filter method to select genes because it is computationally more efficient than the hybrid method [3]. Many filter methods are described as individual gene-ranking methods. They evaluate a gene based on its discriminative power for the target classes without considering its correlations with other genes, which may result in the inclusion of irrelevant and noisy genes in a gene subset for cancer classification. These genes increase the dimensionality of the gene subset and, in turn, affect the classification performance. The filter methods also select a number of genes manually, which causes difficulty in usage, especially for beginner biologists.

Hybrid methods usually provide greater accuracy than filter methods because the genes are selected by considering and optimising correlations among genes. Recently, several hybrid methods based on particle swarm optimisation (PSO) have been proposed to select informative genes from gene expression data [6-9]. PSO is a new evolutionary technique proposed by Kennedy and Eberhart [10], which was motivated by simulations of the social behaviour of organisms such as bird flocking and fish schooling. Alba et al. [6] evaluated a new version of PSO, called geometric PSO, for gene selection. However, the experimental results are less significant because geometric PSO is more about generalising optimisers based on a notion of distance where different distance metrics give rise to different operators with regards to the predefined geometric operators. Shen et al. [7] proposed a hybrid of PSO and tabu search approaches for gene selection. Unfortunately, the results obtained by using the hybrid method are less meaningful because tabu approaches in PSO are unable to search for a near-optimal solution in search spaces. An improved binary PSO was proposed by Chuang et al. [8]. This approach exhibited 100% classification accuracy in many data sets, but it used a high number of selected genes (large gene subset) to achieve the high accuracy. This method uses a large number of genes because the global best particle is reset to the zero position when its fitness values do not change after three consecutive iterations. Li et al. [9] then introduced a hybrid of the PSO and genetic algorithms (GA) for the same purpose. Unfortunately, the accuracy was still not high, and many genes were selected for cancer classification because there were no direct probability relations between GA and PSO. The PSO-based methods are generally intractable to efficiently produce a small (near-optimal) subset of informative genes for high classification accuracy [6-9], mainly because the total number of genes in the gene expression data is too large (high-dimensional data).

One diagnostic goal is to develop a medical procedure based on the least number of genes that are needed to detect diseases. We propose an improved (modified) binary PSO (IPSO) to select a small (near-optimal) subset of informative genes that is most relevant for cancer classification. To test the effectiveness of our proposed method, we applied IPSO to four gene expression data sets, including binary- and multi-class data sets.

This paper is organised as follows. In Section 2, we briefly describe the conventional version of binary PSO and IPSO. Section 3 presents the data sets used and the experimental results. Section 4 summarises this paper by providing its main conclusions and addresses future directions.

2. **Methods.**

2.1. **The conventional version of binary PSO (BPSO).** Binary PSO (BPSO) is initialised with a population of particles. At each iteration, all particles move in a problem space to find the optimal solution. A particle represents a potential solution in an n-dimensional space. Each particle has position and velocity vectors for directing its movement. The position and velocity vectors of the ith particle in the n-dimension can be represented as $X_i = (x_i^1, x_i^2, \ldots, x_i^n)$ and $V_i = (v_i^1, v_i^2, \ldots, v_i^n)$, respectively, where $x_i^d \in \{0, 1\}$; $i = 1, 2, \ldots, m$ (m is the total number of particles) and $d = 1, 2, \ldots, n$ (n is the dimension of data) [11]. v_i^d is a real number for the dth dimension of the particle i, where the maximum v_i^d is $V_{\max} = (1/3) \times n$. This value is important, as it controls the granularity of the search by clamping the escalating velocities. Large values of $V_{\max}$ facilitate global exploration, while small values encourage local exploitation. If $V_{\max}$ is too small, the swarm may not explore sufficiently beyond locally good regions. In addition, $V_{\max}$ increases the number of time steps to reach an optimum and may become trapped in a local optimum. On the other hand, values of $V_{\max}$ that are too large risks missing a good region. The particles may jump over good solutions and continue to search in fruitless regions of the search space. After many tests, we found that an appropriate maximum velocity value is $(1/3) \times n$. We choose $V_{\max}= (1/3) \times n$ and limit the velocity within the range $[1, (1/3) \times n]$ which prevents an overly large velocity. A particle can be near an optimal solution, but a high velocity may make it move far away. By limiting the maximum velocity, particles cannot fly too far away from the optimal solution. The BPSO method has a greater chance of finding the optimal solution under the limit.

In gene selection, the vector of the particle position is represented by a binary bit string of length n, where n is the total number of genes. Each position vector (X_i) denotes a gene subset. If the value of the bit is 1, then the corresponding gene is selected. By contrast, a value of 0 means that the corresponding gene is not selected. Each particle in the tth iteration updates its own position and velocity according to the following equations:

$$\begin{aligned} v_i^d(t+1) &= w(t) \times v_i^d(t) + c_1 r_1^d(t) \times \left(pbest_i^d(t) - x_i^d(t)\right) \\ &\quad + c_2 r_2^d(t) \times \left(gbest^d(t) - x_i^d(t)\right) \end{aligned} \tag{1}$$

$$Sig\left(v_i^d(t+1)\right) = \frac{1}{1 + e^{-v_i^d(t+1)}} \tag{2}$$

$$\textbf{if } Sig\left(v_i^d(t+1)\right) > r_3^d(t), \textbf{ then } x_i^d(t+1) = 1; \textbf{ else } x_i^d(t+1) = 0 \tag{3}$$

where c_1 and c_2 are the acceleration constants in the interval [0, 2] and $r_1^d(t)$, $r_2^d(t)$, $r_3^d(t) \sim U(0, 1)$ are random values in the range $[0, 1]$, which are sampled from a uniform distribution. $Pbest_i(t) = (pbest_i^1(t), pbest_i^2(t), \ldots, pbest_i^n(t))$ and $Gbest(t) = (gbest^1(t), gbest^2(t), \ldots, gbest^n(t))$ represent the best previous position of the ith particle and the global best position of the swarm (all particles), respectively. They are assessed base on a fitness function. $Sig(v_i^d(t+1))$ is a sigmoid function where $Sig(v_i^d(t+1)) \in [0, 1]$. $w(t)$ is an

inertia weight, which was introduced by Shi and Eberhart [12] as a mechanism to control the exploration and exploitation abilities of the swarm and eliminate the need for velocity clamping. The inertia weight controls the momentum of the particle by weighting the contribution of the previous velocity, namely, by controlling how much memory of the previous particle direction will influence the new velocity. In this paper, a nonlinear decreasing approach was applied in BPSO to update $w(t)$ in each iteration. In this approach, an initially large value decreases nonlinearly to a small value. It also allows a shorter exploration time than a linear decreasing approach, with more time spent on refining the solution (exploiting). $w(t)$ was initialised with a value of 1.4 and was updated as follows [13,14]:

$$w(t+1) = \frac{(w(t) - 0.4) \times (MAXITER - Iter(t))}{(MAXITER + 0.4)} \tag{4}$$

where $MAXITER$ is the maximum iteration (generation) and $Iter(t)$ is the current iteration. Figure 1 shows the pseudo code of BPSO.

```
Initialize m particles with n-dimension;
        // m is the total number of particles
REPEAT
  FOR each particle i = 1, . . . , m   DO
    Calculate fitness value
    IF (the fitness value is better than the best fitness
        value (Pbest) in history)     THEN
        Set the current particle position as the new Pbest;
  ENDFOR
  Select the particle with the best fitness value
         of all the particles as the Gbest;
  FOR each particle i = 1, . . . , m   DO
    Update particle velocity according Equation (1)
    Update particle position according Equation (3)
  ENDFOR
UNTIL stopping condition is TRUE
```

FIGURE 1. The pseudo code of BPSO

2.1.1. *Investigating the drawbacks of BPSO and previous PSO-based methods.* Before attempting to propose IPSO as a suitable method, it was prudent to find the limitations of BPSO and previous PSO-based methods [6-9]. This subsection investigates the limitations of these methods by analyzing Equation (2) and Equation (3), which are the most important equations for gene selection in binary spaces. Both of these equations are also implemented in BPSO and the PSO-based methods.

The sigmoid function (Equation (2)) represents a probability for $x_i^d(t)$ to be 0 or 1 ($P\left(x_i^d(t) = 0\right)$ or $P\left(x_i^d(t) = 1\right)$). For example,

if $v_i^d(t) = 0$, **then** $Sig(v_i^d(t) = 0) = 0.5$ **and** $P(x_i^d(t) = 0) = 0.5$.
if $v_i^d(t) < 0$, **then** $Sig(v_i^d(t) < 0) < 0.5$ **and** $P(x_i^d(t) = 0) > 0.5$.
if $v_i^d(t) > 0$, **then** $Sig(v_i^d(t) > 0) > 0.5$ **and** $P(x_i^d(t) = 0) < 0.5$.

In addition, $P(x_i^d(t) = 0) = 1 - P(x_i^d(t) = 1)$. From the analysis, we concluded that $P(x_i^d(t) = 0) = P(x_i^d(t) = 1) = 0.5$ for the initial iteration, because Equation (2) is a standard sigmoid function without any constraint and no modification. Although the next iterations potentially influence the $P(x_i^d(t) = 0)$ or $P(x_i^d(t) = 1)$, the $P(x_i^d(t) = 0) = P(x_i^d(t) = 1) = 0.5$ is almost unchanged for its application on gene expression data because gene expression data is high dimensional and has a large search space. Using this standard sigmoid function in high-dimensional data only reduces the number of genes to

about half of the total number of genes, as shown in the experimental results section. Therefore, Equation (2) and Equation (3) are potentially drawbacks of BPSO and the previous PSO-based methods in selecting a small number of genes for producing a near-optimal (small) subset of genes from gene expression data.

2.2. **An improved (modified) binary PSO (IPSO).** In almost all previous gene expression data research, a subset of genes was selected for excellent cancer classifications. In this article, we propose IPSO for selecting a near-optimal (small) subset of genes in order to overcome the limitations of BPSO and previous PSO-based methods [6-9]. IPSO differs from the BPSO and PSO-based methods in two ways: 1) we introduce a scalar quantity called the particle speed (s), and 2) we propose a rule for updating $x_i^d(t+1)$. By contrast, the BPSO and PSO-based methods use the original rule (Equation (3)) and lack the particle speed implementation. The particles' speed and rule are introduced in order to:

1) increase the probability of $x_i^d(t+1) = 0$ ($P(x_i^d(t+1) = 0)$) and
2) reduce the probability of $x_i^d(t+1) = 1$ ($P(x_i^d(t+1) = 1)$).

The increased and decreased probability values cause a small number of genes to be selected and grouped into a gene subset. $x_i^d(t+1) = 1$ means that the corresponding gene is selected. Otherwise, $x_i^d(t+1) = 0$ indicates that the corresponding gene is not selected.

Definition 2.1. *s_i is the speed, length or magnitude of V_i for the particle i. Therefore, the following properties of s_i are crucial:*

1) non-negativity: $s_i \geq 0$;
2) definiteness: $s_i = 0$ if and only if $V_i = 0$;
3) homogeneity: $\|\alpha V_i\| = \alpha \|V_i\| = \alpha s_i$ where $\alpha \geq 0$;
4) the triangle inequality: $\|V_i + V_{i+1}\| \leq \|V_i\| + \|V_{i+1}\|$ where $\|V_i\| = s_i$ and $\|V_{i+1}\| = s_{i+1}$.

The particles' speed and the rule are proposed as follows:

$$s_i(t+1) = w(t) \times s_i(t) + c_1 r_1(t) \times dist(Pbest_i(t) - X_i(t)) + c_2 r_2(t) \times dist(Gbest(t) - X_i(t)) \quad (5)$$

$$Sig(s_i(t+1)) = \frac{1}{1 + e^{-s_i(t+1)}} \quad (6)$$

subject to $s_i(t+1) \geq 0$

$$\textbf{if } Sig(s_i(t+1)) > r_3^d(t), \textbf{ then } x_i^d(t+1) = 0; \textbf{ else } x_i^d(t+1) = 1, \quad (7)$$

where $s_i(t+1)$ represents the speed of the particle i for the $t+1$ iteration. By contrast, in BPSO and other PSO-based methods (Equations (1)-(3)), $v_i^d(t+1)$ represents a single element of a particle velocity vector for the particle i. In IPSO, Equations (5)-(7) are used to replace Equations (1)-(3), respectively. $s_i(t+1)$ is the rate at which the particle i changes its position. Based on Definition 2.1, the most important property of $s_i(t+1)$ is $s_i(t+1) \geq 0$. Hence, $s_i(t+1)$ is used instead of $v_i^d(t+1)$ so that its positive value can increase $P(x_i^d(t+1) = 0)$.

In Equation (5), $s_i(t+1)$ for each particle is initialised with positive real numbers. The calculation for updating $s_i(t+1)$ is mainly based on the distance between $Pbest_i(t)$ and $X_i(t)(dist(Pbest_i(t) - X_i(t)))$, and the distance between $Gbest(t)$ and $X_i(t)(dist(Gbest(t) - X_i(t)))$, whereas the original formula (Equation (1)) is used to calculate $v_i^d(t+1)$ and is essentially based on the difference between $Pbest_i^d(t)$ and $x_i^d(t)$, and the difference between $Gbest^d(t)$ and $x_i^d(t)$. The distances are used in the calculation for updating $s_i(t+1)$ in order to make sure that Equation (6) is always satisfying the property of $s_i(t+1)$, namely

$(s_i(t+1) \geq 0)$ and to increase $P(x_i^d(t+1) = 0)$. Subsection 2.2.1 explains how to calculate the distance between two positions of two particles, e.g., $dist(Gbest(t) - X_i(t))$.

Equations (5)-(7) and $s_i(t) \geq 0$ increase $P(x_i^d(t) = 0)$ because the minimum value for $P(x_i^d(t) = 0)$ is 0.5 when $s_i(t) = 0$ ($\min P(x_i^d(t) = 0) \geq 0.5$). In addition, they decrease the maximum value for $P(x_i^d(t) = 1)$ to 0.5 ($\max P(x_i^d(t) = 1) \leq 0.5$). Therefore, if $s_i(t) > 0$, then $P(x_i^d(t) = 0) >> 0.5$ and $P(x_i^d(t) = 1) << 0.5$.

Figure 2 shows that a) Equations (5)-(7) and $s_i(t) \geq 0$ in IPSO increase, and b) Equations (1)-(3) in BPSO yield $P(x_i^d(t) = 0) = P(x_i^d(t) = 1) = 0.5$. As an example, the calculations for $P(x_i^d(t) = 0)$ and $P(x_i^d(t) = 1)$ in Figure 2(a) are shown as follows:

if $s_i(t) = 1$, **then** $P(x_i^d(t) = 0) = 0.73$ **and** $P(x_i^d(t) = 1) = 1 - P(x_i^d(t) = 0) = 0.27$.

if $s_i(t) = 2$, **then** $P(x_i^d(t) = 0) = 0.88$ **and** $P(x_i^d(t) = 1) = 1 - P(x_i^d(t) = 0) = 0.12$.

This high probability of $x_i^d(t) = 0$ ($P(x_i^d(t) = 0)$) causes a small number of genes to be selected in order to produce a near-optimal (small) gene subset from high-dimensional data (gene expression data). Hence, IPSO is proposed to overcome the limitations of BPSO and the previous PSO-based methods and to produce a small subset of informative genes.

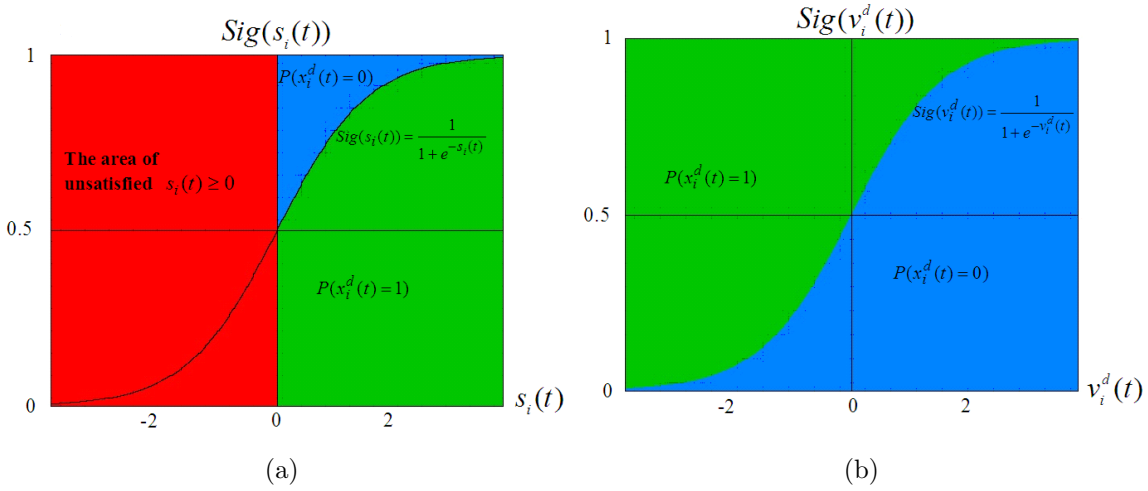

FIGURE 2. The areas of $P(x_i^d(t) = 0)$ and $P(x_i^d(t) = 1)$ based on sigmoid functions in a) IPSO; b) BPSO. The blue and green colors show the areas for $P(x_i^d(t) = 0)$ and $P(x_i^d(t) = 1)$, respectively, and whereas the red color indicates the part of unsatisfied $s_i(t) \geq 0$.

2.2.1. *Calculating the distance of two particles' positions.* The number of different bits between two particles is related to the difference between their positions. For example, $Gbest(t) = [0011101000]$ and $X_i(t) = [1110110100]$. The difference between $Gbest(t)$ and $X_i(t)$ is $diff(Gbest(t) - X_i(t)) = [-1\ -1010\ -11\ -100]$. A value of 1 indicates, this bit (gene) should be selected in comparison with the best position. However, if it is not selected, the classification quality may decrease and lead to a lower fitness value. In contrast, a value of -1 indicates that this bit should not be selected in comparison with the best position, but it is selected. The selection of irrelevant genes makes the length of the subset longer and leads to a lower fitness value. The number of 1 is assumed to be a, whereas the number of -1 is b. We use the absolute value of $a - b$ ($|a - b|$) to express the distance between the two positions. In this example, the distance between $Gbest(t)$ and $X_i(t)$ is $dist(Gbest(t) - X_i(t)) = |a - b| = |2 - 4| = 2$.

2.2.2. *Fitness functions.* The fitness value of a particle (a gene subset) is calculated as follows:

$$fitness(X_i) = w_1 \times A(X_i) + (w_2 (n - R(X_i)) / n) \quad (8)$$

in which $A(X_i) \in [0, 1]$ is the leave-one-out-cross-validation (LOOCV) classification accuracy that uses the only genes in a gene subset (X_i). This accuracy is provided by support vector machine (SVM) classifiers. $R(X_i)$ is the number of selected genes in X_i. n is the total number of genes for each sample. w_1 and w_2 are two priority weights, which correspond to the importance of the accuracy and the number of selected genes, respectively. In this article, the accuracy is more important than the number of selected genes. Therefore, we selected the value of w_1 in the range $[0.6, 0.9]$ and set $w_2 = 1 - w_1$. The value of w_2 was set to $1 - w_1$ to obtain the remaining percentage of weights after the value of w_1 was chosen.

3. **Experiments.**

3.1. **Data sets and experimental setup.** The gene expression data sets used in this study are summarised in Table 1. They included binary- and multi-class data sets that have thousands of genes (high-dimensional data). All of the experimental results reported in this article are acquired using Rocks Linux version 4.2.1 (Cydonia) on the IBM xSeries 335 (cluster node that contains 13 compute nodes). Each compute node has four high performances and 3.0 GHz Intel Xeon CPUs with 512 MB of memory. Thus, a total of 52 CPUs for the 13 compute nodes were used. The IBM xSeries 335 with 52 CPUs was needed to experiment with IPSO and BPSO because both of these methods have huge computational times and run on high-dimensional data. The computational power of IBM xSeries 335 can reduce the computational time of both methods on high-dimensional data. In order to make sure the running time of every run used the same capacity of CPU usage, each run was independently experimented on only one CPU, which was important because the comparison of running times between IPSO and BPSO was used for evaluation of their performances.

The experimental results using IPSO are compared with an experimental method (BPSO) and other PSO-based methods [6-9]. We first applied the gain ratio technique for pre-processing in order to pre-select the top 500-ranked genes, which were then used by IPSO and BPSO. Next, SVM was used to measure the LOOCV accuracy on the gene subsets produced by IPSO and BPSO. In order to avoid selection bias, the LOOCV was implemented in exactly the same way as described by Chuang et al. [8]. Only one

Table 1. The description of gene expression data sets

Data Sets	Number of Samples	Number of Genes	Number of Classes	Source
Leukemia	72	7,129	2	http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi
Lung	181	12,533	2	http://chestsurg.org/publications/2002-microarray.aspx
MLL	72	12,582	3	http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi
SRBCT	83	2,308	4	http://research.nhgri.nih.gov/microarray/Supplement/

Note:
MLL = mixed-lineage leukemia.
SRBCT = small round blue cell tumor.

cross-validation cycle (LOOCV, outer loop) was used and not two nested ones. Several experiments were independently conducted 10 times on each data set using IPSO and BPSO. An average result of the 10 independent runs was obtained. Two criteria were considered to evaluate the performances of IPSO and BPSO: LOOCV accuracy and the number of selected genes. In addition, the running times were also measured for the comparison between IPSO and BPSO. High accuracy, a small number of selected genes, and low running time were needed for an excellent performance. Table 2 contains the parameter values for IPSO and BPSO, which were chosen based on the results of preliminary runs. The numbers of particles and iterations to reach a good solution are problem dependent [15]. If the numbers were large, IBPSO needed more time to complete its process. If the numbers were small, IBPSO took short period of time, but it could not found a good solution. Therefore, we choose intermediate values for the number of particles and iterations betweb 100 and 300. The value of w_1 is larger than w_2 because the classification accuracy is more important than the number of selected genes. We tried to get the best value based on trial and error approaches. IBPSO and BPSO were analyzed using different parameters values. So far, the best values for both w_1 and w_2 using these methods are 0.8 and 0.2, respectively. When w_2 was more than w_1, the number of selected genes also increased. c_1 and c_2 had the same value (2) so that particles were attracted towards the averages of $Pbest_i(t)$ and $Gbest(t)$ [15].

Table 2. Parameter settings for IPSO and BPSO

Parameters	**Values**
The number of particles	100
The number of iteration (generation)	300
w_1	0.8
w_2	0.2
c_1	2
c_2	2

3.2. **Experimental results.** Based on the standard deviation of the classification accuracy shown in Table 3, the results from IPSO were consistent on all data sets. Interestingly, all runs achieved 100% LOOCV accuracy with less than 30 selected genes on all of the data sets. Moreover, the standard deviations of the number of selected genes were less than 1.6 for all of the data sets except for the SRBCT data set (8.32 standard deviations). All of the best results achieved 100% LOOCV accuracy with not more than 6 selected genes, indicating that IPSO efficiently selected and produced a near-optimal gene subset from high-dimensional data (gene expression data).

Practically, the best subset of a data set is first chosen and the genes in it are then listed for biological usage. These informative genes among the thousand of genes may be excellent candidates for clinical and medical investigations. Biologists can save time because they can directly refer to the genes that have higher possibilities of being useful for cancer diagnoses and as drug targets in the future. The best subset is chosen based on the highest classification accuracy with the smallest number of selected genes. The highest accuracy provides confidence for the most accurate classification of cancer types. Moreover, the smallest number of selected genes for cancer classification can reduce the cost in clinical settings.

Figure 3 shows that the averages of the fitness values of IPSO increased dramatically after a few generations on all of the data sets. This trend indicates that IPSO is appropriate for selecting a small number of genes from high-dimensional data (gene expression

TABLE 3. Experimental results for each run using IPSO on leukemia, lung, MLL and SRBCT data sets

Run#	Leukemia		Lung		MLL		SRBCT	
	#Acc (%)	#Selected Genes	#Acc (%)	#Selected Genes	#Acc (%)	#Selected Genes	#Acc (%)	#Selected Genes
1	100	4	100	9	100	7	100	10
2	100	2	100	6	100	6	100	22
3	100	4	100	6	100	7	100	25
4	100	4	100	5	100	6	100	8
5	100	3	100	6	100	8	100	28
6	100	4	100	8	100	4	100	12
7	100	4	100	4	100	5	100	14
8	100	3	100	5	100	7	100	26
9	100	4	100	7	100	8	100	24
10	100	3	100	6	100	9	100	6
Average	100	3.50	**100**	**6.20**	100	6.70	100	17.50
± S.D.	±0	±0.71	**±0**	**±1.48**	±0	±1.50	±0	±8.32

Note: Results of the best subsets shown in shaded cells. A near-optimal subset that produces the highest classification accuracy with the smallest number of genes is selected as the best subset. #Acc and S.D. denote the classification accuracy and the standard deviation, respectively, whereas #Selected Genes and Run# represent the number of selected genes and a run number, respectively.

TABLE 4. Comparative experimental results of IPSO and BPSO

Data	Method / Evaluation	IPSO			BPSO		
		Best	#Ave	S.D	Best	#Ave	S.D
Leukemia	#Acc (%)	100	100	0	98.61	98.61	0
	#Genes	2	3.50	0.71	216	224.70	5.23
	#Time	2.28	2.31	0.02	13.86	13.94	0.03
Lung	#Acc (%)	100	100	0	99.45	99.39	0.18
	#Genes	4	6.20	1.48	219	223.33	4.24
	#Time	8.22	8.31	0.05	110.71	111.07	0.23
MLL	#Acc (%)	100	100	0	97.22	97.22	0
	#Genes	4	6.70	1.50	218	228.11	4.86
	#Time	2.24	2.72	0.25	19.37	19.90	0.35
SRBCT	#Acc (%)	100	100	0	100	100	0
	#Genes	6	17.50	8.32	206	221.30	7.35
	#Time	5.52	5.96	0.39	44.86	44.88	0.01

Note: The best result of each data set shown in shaded cells. It is selected based on the following priority criteria: 1) the highest classification accuracy; 2) the smallest number of selected genes. #Acc and S.D. denote the classification accuracy and the standard deviation, respectively, whereas #Genes and #Ave represent the number of selected genes and an average, respectively. #Time stands for running time in the hour unit.

data) to increase the classification accuracy. A high fitness value is obtained by a combination of a high classification rate and a small number (subset) of selected genes. The condition that the proposed particle speed should always be positive real numbers was started in the initialisation method, and the new rule for updating the particles' positions provoked the early convergence of IPSO. The fitness value of IPSO increased slightly for further generations with all of the data sets, indicating that IPSO explored to find a good solution. By contrast, the averages of the fitness values of BPSO were not improved until the last generation due to $P(x_i^d(t) = 0) = P(x_i^d(t) = 1) = 0.5$.

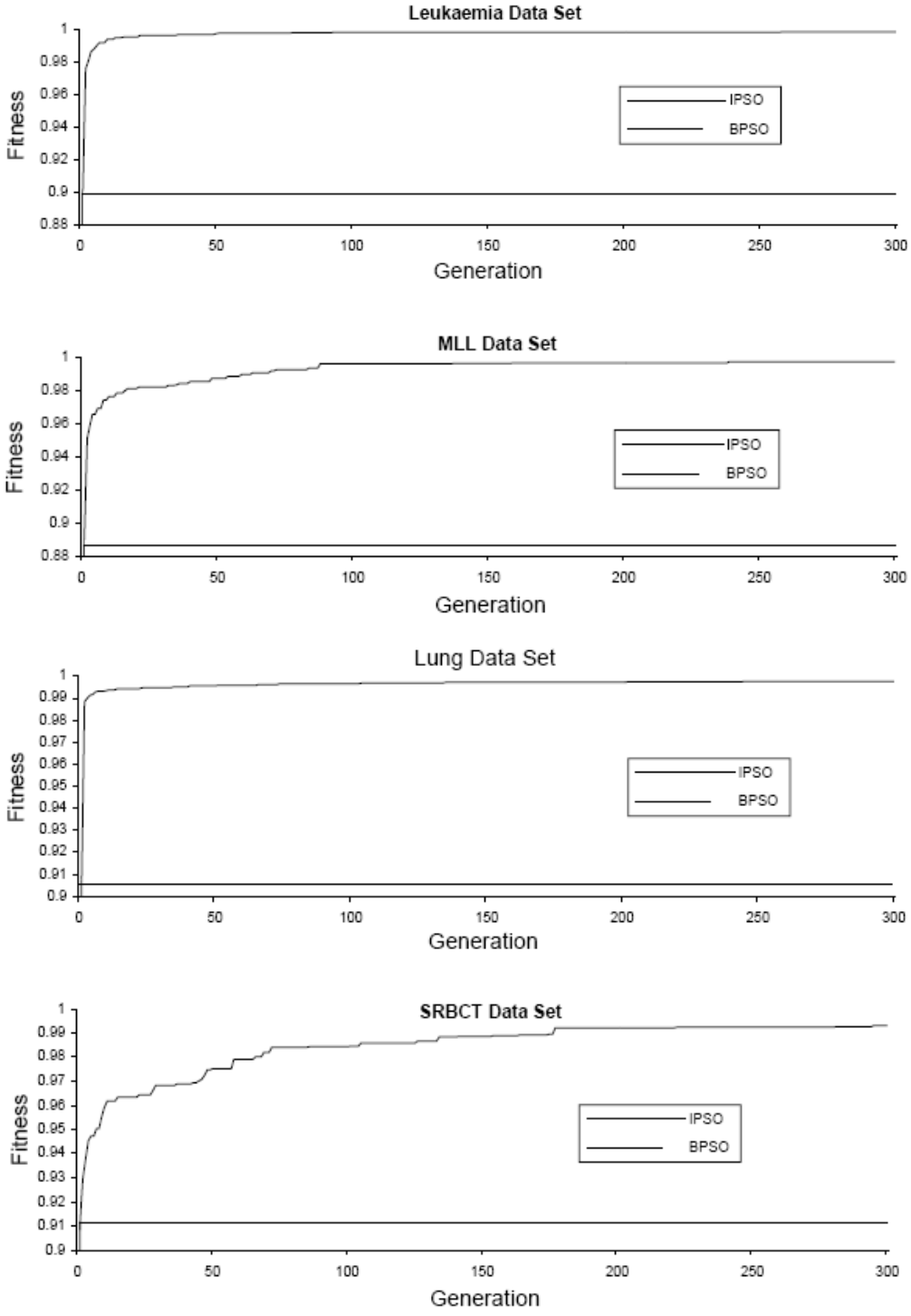

FIGURE 3. The relation between the average of fitness values (10 runs on average) and the number of generations for IPSO and BPSO

As shown in Table 4, the classification accuracy, running time, and the number of selected genes of IPSO were superior to BPSO in terms of the best, average, and standard deviation results on all of the data sets. Moreover, IPSO also produced a smaller number of genes compared to BPSO.

The running times of IPSO were also lower than BPSO in all of the data sets. IPSO can reduce its running times due to the following reasons:

1) IPSO selects a smaller number of genes compared to BPSO;
2) The computation of SVM is fast because it uses a small number of features (genes) that were selected by IPSO for the classification process;
3) IPSO only uses the speed of a particle for comparison with $r_3^d(t)$, whereas BPSO incorporates all elements of a particle's velocity vector for the comparison.

For an objective comparison, we compared our work with previous related works that used PSO-based methods [6-9], as shown in Table 5. Two criteria were used to evaluate the performance of IPSO and the other methods: classification accuracy and the number of selected genes. Higher accuracy with a smaller number of selected genes is needed to obtain superior performance. For all of the data sets except the Lung data set, the averages of the number of the selected genes of our work were smaller [6-9]. Our method also resulted in higher averages of the classification accuracies on all data sets compared to the other methods. The most recent work came up with similar LOOCV results (100%) to ours on Leukemia, MLL, and SRBCT, but they used more than 400 genes to obtain these results [8]. Moreover, they did not have statistically meaningful conclusions because their experimental results were obtained by only one independent run on each data set and not based on average results. The average results are important because their proposed method is a stochastic approach. In additionally, the global best particle position in their approach is reset to the zero position when its fitness values do not change after three successive iterations. Theoretically, their approach is almost impossible to result in a near-optimal gene subset from high-dimensional spaces (high-dimension data) because the global best particles' positions should allow for new exploration and exploitation for finding a near-optimal solution after its position is reset to zero. Overall, our work outperformed the other methods in terms of the LOOCV accuracy and the number of selected genes. The running times between IPSO and these works cannot be compared because they were not reported.

TABLE 5. A comparison between our method (IPSO) and previous PSO-Based methods

Data	**Method** / **Evaluation**	**IPSO**	**IBPSO [8]**	**PSOTS [7]**	**PSOGA [9]**	**GPSO [6]**
Leukemia	#Acc(%)	(100)	100	(98.61)	(95.10)	–
	#Genes	(3.50)	1034	(7)	(21)	–
Lung	#Acc(%)	(100)	–	–	–	(99)
	#Genes	(6.20)	–	–	–	(4)
MLL	#Acc(%)	(100)	100	–	–	–
	#Genes	(6.70)	1292	–	–	–
SRBCT	#Acc(%)	(100)	100	–	–	–
	#Genes	(17.50)	431	–	–	–

Note: The results of the best subsets shown in shaded cells. It is selected based on the following priority criteria: 1) the highest classification accuracy: 2) the smallest number of selected genes. '–' means that a result is not reported in the previous related work. A result in '( )' denotes an average result. #Genes and #Acc represent the number of selected genes and the classification accuracy, respectively.
IBPSO = An improved binary PSO.
PSOGA = A hybrid of PSO and GA.
PSOTS = A hybrid of PSO and tabu search.
GPSO = Geometric PSO.

According to Figure 3 and Tables 3-5, IPSO is reliable for gene selection because it produced the near-optimal solution from the gene expression data, due to the proposed particle speed and the introduced rule that increased the probability $x_i^d(t+1) = 0$ ($P(x_i^d(t+1) = 0)$). The particle speed was introduced to provide the rate at which a particle changes its position, whereas the rule was proposed to update the particle positions. The increased probability value for $x_i^d(t+1) = 0$ causes the selection of a small number of informative genes and produces a near-optimal subset (a small subset of informative genes with high classification accuracy) for cancer classification.

4. **Conclusion.** In this paper, IPSO was proposed for gene selection on four gene expression data sets. Overall, based on the experimental results, the performance of IPSO was superior to BPSO and PSO-based methods in terms of the classification accuracy and the number of selected genes. IPSO was excellent because the probability $x_i^d(t+1) = 0$ was increased by the particle speed and the introduced rule, which were proposed to yield a near-optimal subset of genes for better cancer classification. IPSO also features lower running times because it selects a smaller number of genes compared with BPSO. In future work, a modified representation of the particle positions in PSO will be proposed to reduce the number of gene subsets in solution spaces.

Acknowledgements. The authors thank the referees for helpful suggestions and thank Universiti Teknologi Malaysia for sponsoring this research with a GUP Research Grant (Vot Number: Q.J130000.7107.01H29).

REFERENCES

[1] M. F. Misman, S. Deris, M. S. Mohamad and S. Z. M. Hashim, Identification of significant phenotypes related genes and biological pathways using a hybrid of support vector machines and smoothly clipped absolute deviation, *ICIC Express Letters, Part B: Applications*, vol.1, no.2, pp.131-136, 2010.

[2] S. Knudsen, *A Biologist's Guide to Analysis of DNA Microarray Data*, John Wiley & Sons, New York, USA, 2002.

[3] M. S. Mohamad, S. Omatu, M. Yosioka and S. Deris, A cyclic hybrid method to select a smaller subset of informative genes for cancer classification, *Internztional Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2189-2202, 2009.

[4] M. S. Mohamad, S. Omatu, M. Yoshioka and S. Deris, A three-stage method to select informative genes for cancer classification, *International Journal of Innovative Computing, Information and Control*, vol.6, no.1, pp.117-125, 2010.

[5] M. S. Mohamad, S. Omatu, M. Yoshioka and S. Deris, A two-stage method to select a smaller subset of informative genes for cancer classification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(A), pp.2959-2968, 2009.

[6] E. Alba, J. Garcia-Nieto, L. Jourdan and E. Talbi, Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, *Proc. of IEEE Congress on Evolutionary Computation*, Singapore, pp.284-290, 2007.

[7] Q. Shen, W. M. Shi and W. Kong, Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, *Comput. Biol. Chem.*, vol.32, pp.53-60, 2009.

[8] L. Y. Chuang, H. W. Chang, C. J. Tu and C. H. Yang, Improved binary PSO for feature selection using gene expression data, *Comput. Biol. Chem.*, vol.32, pp.29-38, 2009.

[9] S. Li, X. Wu and M. Tan, Gene selection using hybrid particle swarm optimization and genetic algorithm, *Soft Computing*, vol.12, pp.1039-1048, 2008.

[10] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE Int. Conf. Neural Networks 4*, Perth, Australia, pp.1942-1948, 1995.

[11] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, *Proc. of IEEE Int. Conf. Systems, Man, and Cybernetics*, Florida, USA, vol.5, pp.4104-4108, 1997.

[12] Y. Shi and R. C. Eberhart, A modified particles swarm optimizer, *Proc. of IEEE Congress on Evolutionary Computation*, Piscataway, NJ, pp.69-73, 1998.

[13] S. Naka, T. Genji, T. Yura and Y. Fukuyama, Practical distribution state estimation using hybrid particle swarm optimization, *Proc. IEEE Power Engineering Society Winter Meeting*, Ohio, USA, pp.815-820, 2001.

[14] T. Peram, K. Veeramacheneni and C. K. Mohan, Fitness-distance-ratio based particle swarm optimization, *Proc. of IEEE Swarm Intelligence Symposium*, Indiana, USA, pp.174-181, 2003.

[15] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley and Sons, West Succex, England, 2005.