ISSN: 2788-7243 ISSN: 2788-7235

# Binary Spotted Hyena Optimization Algorithm with Particle Swarm Optimization for Feature Selection

**Jomana Yousef Khaseeb[**] [*]   Rasha Assaf

## Abstract

Feature selection is the process of removing duplicated and unimportant features from the dataset in order to enhance the learning algorithm and data mining. In this paper, a hybrid form of optimization algorithm as a wrapper feature selection model is used to reduce the number of features while trying to improve the classification accuracy. This hybrid algorithm used a binary variant of both Particle Swarm Optimization (PSO) and Spotted Hyena Optimization Algorithm (SHO), which is called BSHPSO. The proposed approach has been tested on five high-dimensional low-instances medical datasets from UCI. The results showed superior performance for the BSHPSO over the binary versions of both PSO and SHO algorithms.

***Keywords:*** Hybrid Algorithm, Binary Optimization Algorithm, Transfer Function, Feature Selection, Particle Optimization Algorithm, Spotted Hyena Optimization

[*] *Applied Computing Dept., Palestine Technical University, Palestine*
[**] *Palestine Polytechnic University*

ISSN: 2788-7243 ISSN: 2788-7235

## Introduction

Feature Selection (FS) is an important task for machine learning algorithms. It is utilized to reduce the search space by eliminating unnecessary features or attributes. FS is a type of optimization problem due to its task of searching the most related features in the dataset without losing any information [1]. Three search strategies are used with FS [1]: complete search that creates and assesses all feasible subsets of features to pick the ideal set. Random search that generates random feature subsets. Heuristic search creates a random subset of features and using heuristic techniques to lead the searching process towards the optimal feature subset.

Two general models were used with FS approaches to determine how to assess the generated feature subsets. These models were generally categorized into filter and wrapper models [2]. Filter models evaluate a feature subset based on the interconnections between the features without using any learning algorithm. While using the feedback of the learning algorithm as the evaluation criteria is associated with the wrapper models. Furthermore, meta-heuristic algorithms would be used more frequently to solve optimization problems. These algorithms mimic the social and organizational behaviors of various creatures in nature such as birds, ants, fish, wolves, and whales. Examples of meta-heuristics: Genetic Algorithm (GA) [3], Dragonfly Algorithm (DA) [4], Honey Bee Mating Optimization (HBMO) [5], Particle Swarm Optimization (PSO) [6], Ant Colony Optimization (ACO) [7], Harris Hawks Optimizer (HHO) [8], Ant Lion Optimizer (ALO) [9], Grey Wolf Optimizer (GWO) [10], Bacterial Foraging Optimization (BFO) [11], Whale Optimization Algorithm (WOA) [12], Spotted Hyena Optimization (SHO) [13], and many others.

In meta-heuristic algorithms, the searching process is divided into two stages: exploration and exploitation. Exploration means a global search, while exploitation means a local search. A good balance between these two stages means enhancing the algorithm’s ability to prevent trapping in the local optima. In this paper, a binary version of a hybrid meta-heuristic algorithm is proposed as a wrapper method used to resolve the FS problem in order to reduce the datasets’ dimension without reducing the classification accuracy. This hybrid algorithm takes advantage of the PSO exploration and the SHO exploitation to improve the search process for feasible features in a reasonable

ISSN: 2788-7243 ISSN: 2788-7235

time. Transfer functions are [14] used in this paper to convert the solutions into binary format.

The proposed approach was tested on high-dimensional and low instances medical datasets. The KNN classifier was used to evaluate this approach. The obtained results proved that the BSHPSO model obtained better results than the original PSO and SHO. In addition to the best results than the results of the well-known approaches. The results indicate that BSHPSO outperforms other optimizers in classification accuracy, the number of selected features, and fitness values.

The rest of this paper is organized as follows. Section 2 briefly describes PSO, SHO, and the binary version of optimization algorithms. The proposed approach is described in Section 3. The datasets and the experimental setup are presented in Section 4. Section 5 discusses the experimental results. Finally, the conclusion is summarized in Section 6.

## Background

### *Particle Swarm Optimization PSO*

It was developed by Kennedy and Eberhart [15]. PSO emulates the socio-behavioral interactions that occur between particles, such as the birds in the swarm while they are searching for food. Each particle in the swarm represents a single solution in the search space. The particles update their positions depending on the best position obtained by the whole swarm, in addition to the best position of the particle itself. At the beginning of the algorithm, the population's velocity and position are initialized randomly. The velocities and the positions are updated according to Eq. 1 and Eq. 2.

$$vl^n = wt * vl^o + acc1 * rand1 * (pbest - ps^o) + acc2 * rand2 * (gbest - ps^o) \quad (1)$$

where $vl^o$ is the particle's old velocity. $ps^o$ is a particle old position, *rand1* and *rand2* are two randoms in the interval [0,1]. *acc1* and *acc2* are acceleration factors that represent a particle's dependent on *pbest* and *gbest*. The solutions *pbest* and *gbest* are the personal best and the global best, respectively. *pbest* is the best solution for the particle itself while *gbest* is a global best of the whole swarm. *wt* is an inertia weight that is used to balance the exploration and exploitation aspects during the search process. It decreased linearly or non-linearly according to the equations presented in [16, 17].

ISSN: 2788-7243 ISSN: 2788-7235

A new position, $ps^n$, is calculated by the following equation:

$$ps^n = vl^n + ps^o \tag{2}$$

*Spotted Hyena Optimization Algorithm SHO*

The spotted hyena optimization introduced by G. Dhiman and V. Kumar in [18], which emulates the hunting attitudes of spotted hyenas. SHO starts by generating the population randomly, then the fitness value for each solution is calculated using the appropriate objective function that is suitable for the optimization problem. The best solution is designated as the prey, and the other search agents will change their positions based on the prey's position. This simulates the procedure of encircling the prey and shrinking towards it. This can be modelled in Eq. 3 and Eq. 4:

$$Ds = |B \cdot XP(t) - XS(t)| \tag{3}$$

$$XS(t+1) = XP(t) - E \cdot Ds \tag{4}$$

where $t$ and $t$+1 present the current and next iteration, $XP$ and $XS$ indicate the position of the prey and other spotted hyenas, respectively. $B$ and $E$ are two coefficients that are calculated using Eq. 5 and Eq. 6.

$$B = 2 \cdot rand1 \tag{5}$$

$$E = 2h \cdot rand2 - h \tag{6}$$

where *rand1* and *rand2* are random numbers between 0 and 1. Vector $h$ is calculated based on the Eq. 7.

$$h = 5 - \left(\frac{5 * iter}{ITERATIONS}\right) \tag{7}$$

where $h$ is a vector that decreases linearly from 5 to 0 to balance between the exploration and exploitation phases during the iterations. *iter* is the current iteration, and *ITERATIONS* is the maximum number of iterations in the algorithm.

The SHO determines the optimal solutions to modify the locations of other search agents. This mimics the hunting mechanism, which is described by the following equations:

$$Ds_h = |B \cdot X_h - X_k| \tag{8}$$

$$X_k = X_h - E \cdot Ds_h \tag{9}$$

ISSN: 2788-7243 ISSN: 2788-7235

**(10)** $$Ch = Xk + Xk{+}1 + \dots + Xk{+}M$$

where $X_h$ defines the position of first best spotted hyena, $X_k$ is the position of other agents, $B$ and $E$ are coefficients calculated as in Eq. 5 and Eq. 6. $M$ defines the number of spotted hyenas which is calculated as follows:

**(11)** $$M = count(X_h, X_{h+1}, X_{h+2}, \dots, X_h + N)$$

where N is a random number in [0.5, 1], and M is number of optimal solutions. After Encircling the prey, the spotted hyenas start to attack the prey. The mathematical formula that simulate the attack behavior is described by the following equation:

$$X_{t+1} = \frac{C_h}{M}$$

**(12)**

where $X_{t+1}$ is the best solution that updates the positions of other search agents.

*Binary Optimization Problem*

FS is a kind of binary optimization problem since the features can be represented as 0 for non-selected features and 1 for selected ones. Meta-heuristics can be used in the literature to resolve the optimization problems for continuous and binary search spaces. One of the well-known methods that are used to switch the continuous solutions into binaries is the use of transfer functions (TF) [14]. S-shaped and V-shaped are two types of TF that are used for the conversion issue. These functions produce a probability to convert the binary solution from 0 to 1 and vice versa.

Kennedy and Eberhart in [19] and Rashedi et al. in [20] presented Eq. 13 and Eq. 14, respectively, to flip the binary bits from zero to one and conversely.

(13) $$bin_{t+1} = \begin{cases} 1 & if\ Transfer(val_{t+1}) > rand \\ 0 & Otherwise \end{cases}$$

where $bin_{t+1}$ is a binary value at iteration $t + 1$. $Transfer(val_{t+1})$ is the probability value that can be retrieved by any transfer function to flip a binary bit. rand is a random number in the interval [0,1].

(14) $$bin_{(t+1)} = \begin{cases} Complement(bin_{(t)}) & ,if\ Transfer(val_{(t+1)}) > rand \\ bin_{(t)} & ,Otherwise \end{cases}$$

ISSN: 2788-7243 ISSN: 2788-7235

where $bin_{(t)}$ and $bin_{(t+1)}$ are the binary solutions at iterations $t$ and $t$ +1, respectively. The function Complement() is the complement of any binary solution. *Transfer*() is a transfer function of the value *val*($t$ +1), that retrieves the probability to covert the binary bits of any solution. *rand* is any random value between 0 and 1.

## The proposed approach

This paper proposed a binary hybrid meta-heuristic algorithm that is utilized as a wrapper FS method. This method seeks to find reasonable features from the dataset that are beneficial for the classification task. The proposed approach would have to prevent the local optima and achieve the global optima by adjusting the exploitation and exploration aspects during the search process. A binary form of both PSO and SHO is used to design a hybrid approach that interests the exploration and exploitation ability of both PSO and SHO, respectively. This work used sigmoid function in eq. 15 which is one of S-shaped TF to modify continuous solutions into binary ones.

$$S(x) = \frac{1}{1+e^{-x}} \tag{15}$$

Eq. 16 is the objective function that is used to assess each candidate solution. This function joins the size of the selected attributes and the classification accuracy. The best solution is the one that has the lowest objective function, which indicates the lowest number of selected attributes and the best classification accuracy.

$$ObjectiveFunction = pErr + q\frac{|n|}{|N|} \tag{16}$$

where *Err* is a classification error. |*n*| is the size of selected attributes and |*N*| is the overall set of attributes. The two parameters $p$ and $q$ are the importance of the classifier performance and the number of picked attributes. $p$ is any value within the interval [0,1] and $q = (1-p)$ as mentioned in [21].
SHO algorithm may trap in a local optimum. BSHPSO combines both PSO and SHO to enhance the global search capability when dealing with high-dimensional datasets. This approach is proposed in more detail in the following subsection.

ISSN: 2788-7243 ISSN: 2788-7235

***Binary Spotted Hyena Optimization with Particle Swarm Optimization BSHPSO***

In this approach, the BPSO algorithm is used to enhance the exploration aspect of BSHO. As mentioned earlier, PSO has a strong exploration ability while SHO has a strong exploitation ability. The binary form of a hybrid approach can protect BSHO from dropping into a local optimum by depending on the BPSO's superior exploration ability. This approach is called the hybrid Spotted Hyena Optimization Algorithm with Particle Swarm Optimization Algorithm (BSHPSO).

The initial population for the BSHO algorithm is generated randomly. BSHO improves the solutions then passes them to the BPSO algorithm. The BPSO also improves the solutions again and returns them to the BSHO algorithm. These processes will be iterated until stopping criteria are fulfilled. Algorithm 1 shows the pseudo-code of the hybrid algorithm. In lines 2-6, the population size is determined, the solutions are generated randomly, and all the parameters are initialized. Then the fitness value of each solution is calculated, and the best solution is determined as shown in lines 7-9. Each spotted hyenas updates its position according to the BSHO equations as illustrated during line 14. The new fitness values are calculated in line 17 and gbest is identified in line 18. The new velocities and locations are calculated according to the BPSO equations as shown in lines 22 and 23.

The Datasets and the Experimental Setup

*The Datasets*

Five UCI datasets [22] are used to test the experiments. The datasets are presented in Table 1 along with their identifiers, such as the number of instances, attributes, and classes. These datasets are high-dimensional small-instance datasets with thousands or millions of features and a limited number of instances. Dealing with these datasets is tricky since a limited number of samples is insufficient to train the learning model. Furthermore, the large number of features extends the search field and raises the computational complexity.

**Algorithm 1 Hybrid Spotted Hyena with Particle Swarm Optimization BSHPSO Algorithm**

1: Start

2: Set population size P (No. of search agents)

ISSN: 2788-7243 ISSN: 2788-7235

```
Set number of iterations T
Initialize the random positions and velocities of the search agents
Initialize SHO parameters h, B, E, M
Initialize PSO parameters c1, c2, and w
Calculate the fitness of each search agent
Find the best solution X_h
Find the group of optimal solutions C_h
t ← 1
while t < T do
    p ← 1
    for p < P do
        Update the position of each search agent according to Eqs.12
    end for
    Update h, B, E, M according to Eqs. 7, 5, 6, and 11
    Calculate the new fitness of each search agent
    Find global best X_h
    Update the group of optimal solutions C_h
    for p < P do
        Find personal best bpest
        Calculate the new velocity for each particle based on PSO using Eq. 1
        Find the new binary position based on Eq. 13 and Eq. 15
    end for
    t++
end while
Return the best solution X_h
End
```

**Table 1: Summary of high dimensional low samples data sets**

| Data set | No. of samples | No. of features | No. of classes |
|---:|---:|---:|---:|
| 11_Tumors | 174 | 12533 | 11 |
| 14_Tumors | 308 | 15009 | 26 |
| Leukemia1 | 72 | 5327 | 3 |
| Leukemia2 | 72 | 11225 | 3 |
| Prostate_Tumor | 102 | 10509 | 2 |

ISSN: 2788-7243 ISSN: 2788-7235

## Experimental Setup

According to a cross-validation manner, records in the datasets are split randomly into 80% training and 20% testing subsets. The implementations were done using MATLAB 2013 and the algorithms were run 20 times on Intel Core i5, 2.2 GHz CPU, and 4GB RAM. The results are recorded upon 100 iterations with a population size equal to 10. In the fitness equation, parameters $p$ and $q$ are assigned to 0.99 and 0.01, respectively.

## Evaluation Metrics

The evaluation metrics that are used to evaluate the optimization algorithms are average classification accuracy, average FS size, average fitness value, and average running time. The average classification accuracy is defined by the following equation:

$$AvgAccuracy = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{N}\sum_{j=1}^{N}(Pn_j = Al_j) \quad (17)$$

where $T$ is the total run, $N$ is the number of instances in the dataset, $Pn$ and $Al$ are the predictive and actual class, respectively.

The average number of selected features is defined by the following equation:

$$AvgSelectedFeatures = \frac{1}{T}\sum_{i=1}^{T}\frac{s_i}{S} \quad (18)$$

where $T$ is the total run, $s_i$ is the best number of features and $S$ is the overall features. The average fitness value is by the following equation:

$$AvgFitness = \frac{1}{T}\sum_{i=1}^{T}Fitness_i \quad (19)$$

where $T$ is the total run and $Fitness_i$ the best fitness value.

The average running time is defined by the following equation:

$$AvgTime = \frac{1}{T}\sum_{i=1}^{T}Time_i \quad (20)$$

where $T$ is the total run and $Time_i$ the running time at iteration $i$.

ISSN: 2788-7243 ISSN: 2788-7235

Experimental Results and Discussion

Several observations are taken in order to get a broad view of the advantages and drawbacks of the binary hybrid solution (BSHPSO). To convert continuous values into binaries, the S-shaped TF was being used. The BSHPSO method was compared with the native versions of BSHO and BPSO based on the S-shaped TF.

Tables 2, 3, 4, and 5 show the outcomes of the comparisons between BSHO, BPSO, and BSHPSO based on S-shaped TF according to the classification precisions, number of picked features, fitness results, and running time. Table 2 reveals that BSHPSO had the best accuracy on four datasets, whereas BPSO had the best accuracy on one. With regards to classification accuracy, BSHO had been unable to compete with other methods.

BSHPSO achieved the best accuracy performance by selecting the smallest features on all datasets, according to the results in table 3. BSHO and BPSO were unable to surpass the proposed method in terms of selecting appropriate features.

The results based on the fitness values are given in 4. The findings showed BSHPSO was the best classifier on four datasets. In the case of BPSO, the best results were obtained on one dataset.

The results are shown in Table 5 based on the average running time. The best running time was observed by BPSO followed by BSHO then BSHPSO. The results conclude that BSHPSO was the best according to the fitness values that combine the lowest possible features with the finest classification accuracy. In other words, BSHPSO is the best S-shaped TF approach.

**Table 2: Comparison of the original BSHO and BPSO with BSHPSO based on S-shaped TF in terms of**

| Dataset | Metric | BSHO | BPSO | BSHPSO |
|---|---|---|---|---|
| 11_Tumors | Avg | 0.7822 | 0.8055 | 0.8375 |
| 14_Tumors | Avg | 0.6011 | 0.5999 | 0.6913 |
| Leukemia1 | Avg | 0.8733 | 0.9322 | 0.9474 |
| Leukemia2 | Avg | 0.8223 | 0.7422 | 0.8685 |
| Prostate_Tumor | Avg | 0.8944 | 0.9524 | 0.9413 |

ISSN: 2788-7243 ISSN: 2788-7235

**Table 3: Comparison of the original BSHO and BPSO with BSHPSO based on S-shaped TF in terms of number of selected features**

| Dataset | Metric | BSHO | BPSO | BSHPSO |
|---|---|---|---|---|
| 11_Tumors | Avg | 7530.28 | 6315.57 | 2077.15 |
| 14_Tumors | Avg | 9630.35 | 7500.13 | 3300.62 |
| Leukemia1 | Avg | 3320.22 | 2563.80 | 757.52 |
| Leukemia2 | Avg | 5633.78 | 5392.90 | 1519.94 |
| Prostate_Tumor | Avg | 6411.63 | 5112.15 | 1539.27 |

**Table 4: Comparison of the original BSHO and BPSO with BSHPSO based on S-shaped TF in terms of fitness results**

| Dataset | Metric | BSHO | BPSO | BSHPSO |
|---|---|---|---|---|
| 11_Tumors | Avg | 0.2332 | 0.1985 | 0.1738 |
| 14_Tumors | Avg | 0.4033 | 0.4011 | 0.3153 |
| Leukemia1 | Avg | 0.1401 | 0.0794 | 0.0606 |
| Leukemia2 | Avg | 0.2027 | 0.2695 | 0.1352 |

**Table 5: Comparison of the original BSHO and BPSO with BSHPSO based on S-shaped TF in terms of running time in millisecond**

| Dataset | Metric | BSHO | BPSO | BSHPSO |
|---|---|---|---|---|
| 11_Tumors | Avg | 180.3166 | 136.3661 | 347.4319 |
| 14_Tumors | Avg | 748.0990 | 441.8360 | 1338.7338 |
| Leukemia1 | Avg | 18.5398 | 14.4950 | 41.0150 |
| Leukemia2 | Avg | 36.2015 | 27.7031 | 97.8016 |
| Prostate_Tumor | Avg | 57.8930 | 40.6901 | 134.5022 |

## CONCLUSION

This paper presented the BSHPSO hybrid algorithm, which was used to solve the FS issue. This hybrid method takes the benefits of both BSHO and BPSO optimizers. The hybrid method was tested using the assessment metrics on five UCI high-dimensional smallinstance datasets. The comparisons of BSHPSO, BPSO, and BSHO based on S-shaped TFs revealed that BSHPSO

ISSN: 2788-7235 ISSN: 2788-7243

is the best approach across all assessment metrics. The BSHPSO approach's impressive results showed its ability to adapt the behavior of exploitation and exploration through iterations.

Future research should consider combining the BSHO algorithm with other meta-heuristic algorithms. Furthermore, rather than using UCI datasets, it would be interesting to use the BSHPSO approach to solve real datasets.

ISSN: 2788-7243 ISSN: 2788-7235

## References

A. A. Heidari and R. A. Abbaspour, "Enhanced chaotic grey wolf optimizer for real-world optimization problems: A comparative study," in *Handbook of Research on Emergent Applications of Optimization Algorithms*, pp. 693–727, IGI Global, 2018.

A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849–872, 2019.

C. Blake, "Uci repository of machine learning databases," *http://www. ics. uci. edu/˜ mlearn/MLRepository. html*, 1998.

C. Yang, W. Gao, N. Liu, and C. Song, "Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight," *Applied Soft Computing*, vol. 29, pp. 386–394, 2015.

E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Bgsa: binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.

G. Dhiman and V. Kumar, "Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.

H. Jia, J. Li, W. Song, X. Peng, C. Lang, and Y. Li, "Spotted hyena optimization algorithm with simulated annealing for feature selection," *IEEE Access*, vol. 7, pp. 71943– 71962, 2019.

H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*, vol. 454. Springer Science & Business Media, 2012.

J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 5, pp. 4104–4108, IEEE, 1997.

ISSN: 2788-7243 ISSN: 2788-7235
J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

K. Chen, F.-Y. Zhou, and X.-F. Yuan, “Hybrid particle swarm optimization with spiralshaped mechanism for feature selection,” *Expert Systems with Applications*, vol. 128, pp. 140–156, 2019.

M. M. Mafarja and S. Mirjalili, “Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection,” *Soft Computing*, vol. 23, no. 15, pp. 6249–6265, 2019.

M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, 2017.

M. Mafarja and S. Mirjalili, “Whale optimization approaches for wrapper feature selection,” *Applied Soft Computing*, vol. 62, pp. 441–453, 2018.

O. B. Haddad, A. Afshar, and M. A. Marino, “Honey-bees mating optimization (hbmo) algorithm: a new heuristic approach for water resources optimization,” *water resources management*, vol. 20, no. 5, pp. 661–680, 2006.

S. Kashef and H. Nezamabadi-pour, “An advanced aco algorithm for feature subset selection,” *Neurocomputing*, vol. 147, pp. 271–279, 2015.

S. Mirjalili and A. Lewis, “S-shaped versus v-shaped transfer functions for binary particle swarm optimization,” *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.

S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

S.-K. S. Fan and Y.-Y. Chiu, “A decreasing inertia weight particle swarm optimizer,” *Engineering Optimization*, vol. 39, no. 2, pp. 203–228, 2007.

Y.-P. Chen, Y. Li, G. Wang, Y.-F. Zheng, Q. Xu, J.-H. Fan, and X.-T. Cui, “A novel bacterial foraging optimization algorithm for feature selection,” *Expert Systems with Applications*, vol. 83, pp. 1–17, 2017.

ISSN: 2788-7235 ISSN: 2788-7243

Z. Zhu, Y.-S. Ong, and M. Dash, “Wrapper–filter feature selection algorithm using a memetic framework,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 70–76, 2007.