Discrete Interval Binary Signal Optimization Using PSO

Sergio Fernandez[1] and Rafael Stubs Parpinelli[2]

[1]Graduate Programm in Electrical Engineering, State University of Santa Catarina, Joinville, Brazil
[2]Graduate Programm in Applied Computing, State University of Santa Catarina, Joinville, Brazil

sfs325@gmail.com, rafael.parpinelli@udesc.br

Abstract. Discrete Interval Binary Signals (DIBS) is a well-known multi-frequency two-level amplitude signal used in system identification and commonly occurs harmonic distortion in the frequency spectrum as a drawback. Hence, an optimization must be done to improve the relation between signal and undesired harmonic distortion to improve the Crest Factor (CR). This paper proposes the use of Particle Swarm Optimization (PSO) to optimize DIBS. The employed PSO is the canonical codified in the continuous domain. Eight different experiments with different parameters of DIBS are proposed to validate this PSO application. PSO performance is compared with the Van Den Bos algorithm widely used in the literature. Results show better performance for PSO compared to the Van Den Bos algorithm passing the Wilcoxon rank sum test with a very small p-value. Another remarkable fact is that PSO shows good convergence to all problem instances, giving robustness to the solution.

Keywords: System identification, PSO, Signal Optimization, Swarm Intelligence

1 Introduction

Frequency domain system identification is the area which tries to estimate the frequency transfer function of black box systems. The system is excited using an input signal and the output response is measured to identify the relation of the output over the input [1]. In frequency domain system identification there are many different types of multi-frequency signals which can be used as input. Some of the most popular are multi-sine, DIBS, maximum length binary sequences (MLBS), or chirp. There are studies evaluating the quality of these signals and DIBS shows superior characteristics over the rest [1–3]. An example where these multi-frequency signals are widely used as input is bioimpedance, allowing fast measurements in multiple frequencies [2, 4, 5].

DIBS is a two-level and periodic signal defined by a set of Fourier coefficients of a Fourier serie. The most important drawback of two-level signals is the production undesired harmonics along the whole frequency spectrum increasing the harmonic distortion and reducing the power on the desired Fourier coefficients reducing the crest factor [6]. In frequency domain system identification there are two conditions to attend: the input signal should be small enough to minimize non-linear distortion and large enough to minimize noise effects [7]. Because of these two characteristics, optimizing the effective power of excitation signals has become a critical and important task in system identification [1, 8, 9].

The optimization of the power of the desired Fourier coefficients in DIBS is achieved through optimizing the relation between the desired Fourier coefficients and the harmonic distortion. It does not exist any numerical method to optimize DIBS, so alternative strategies must be adopted in order to solve this problem. Nowadays, only one approximative method has been described in literature for optimizing DIBS, named Van Den Bos algorithm[10].

Heuristic algorithms are well known for their capability to find good solutions with affordable computational capacity for problems where parameter optimization is needed. Some of them are based on populations such as Genetic Algorithms (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO) [11]. They have been widely used in the area of system identification[12–15] and with the objective of optimizing excitation signals such as multisine[15, 16].

The objective of this paper is to optimize the generation of DIBS signals using PSO and compare it with the Van Den Bos algorithm [10]. To statistically validate the improvement of the solution the non-parametric statistical Wilcoxon signed-rank test is used. P-value is given as two-tailed. The used PSO is the originally proposed by[17] with the particles codified in the continuous domain.

The paper is organized as follows. Section 2 describes the mathematical nature of DIBS. Section 3 exposes the original Van Den Bos algorithm. Section 4 exposes the application of PSO for optimizing DIBS. Section 5 defines the set of experiments and the setup used to evaluate the application of PSO compared with the Van Den Bos algorithm. Section 6 shows the results and analysis. Section 7 is the conclusion and future projects for this application.

2 Discrete Interval Binary Signal Optimization Problem

DIBS are periodic discrete and two-level amplitude signals composed of N number of points. Two-level amplitude signals such as square wave or in this case DIBS, create plenty of undesired harmonics along the frequency spectrum producing harmonic distortion. This results in a reduction of power in the desired Fouirier coefficients with the same amplitude increasing the crest factor.

In order to measure the optimization quality of a signal, some metric has to be used. Focusing on the objective of reducing the harmonic distortion and increasing the signal power an objective P can be defined. P is the relation between the power on the desired Fourier coefficients and the sum of the Fourier coefficients in th whole frequency spectrum [1]. This is measured as follows:

$$P = \frac{\sum_{k=1}^{n} |H_k|}{\sum_{k=1}^{n} |H_{sk}|} \tag{1}$$

where $|H_k|$ are the magnitudes of every Fourier coefficients in the obtained signal (desired signal + undesired harmonic distortion), $|H_{sk}|$ are the magnitudes of the desired Fourier coefficients in the obtained signal (desired signal) and n is the number of Fourier coefficients of the spectrum, equal to $N/2$. The objective of the optimization will be reducing the value of P in order to reduce the harmonic distortion. The theoretical minimum of this metric will be 1 when $|H_k| = |H_{sk}|$, in other words, the harmonic distortion is equal to 0 and the whole power of the signal is on the desired Fourier coefficients. In this signal, because of its binary nature, it is impossible reduce P down to 1 but algorithms can be used to reduce P as much as possible [10].

3 Van Den Bos algorithm for optimizing DIBS

The algorithm was firstly proposed by Van Den Bos in 1979 [10]. It is described in Algorithm 1. It is iterative and convergence is expected in 2-15 iterations [10].

Algorithm 1 Basic DIBS construction

1: New $\theta_{sk(current)} = \{x | x \sim \mathcal{U}(0, 2\pi)\}$
2: Loop is set to true
3: **repeat**
4: $\quad D_k = |D_k| e^{j\theta_{k(current)}}$ Containing $|D_{sk}|$
5: $\quad d_k = \mathcal{F}^{-1} D_k$
6: $\quad h_k = sng(d_k)$; If $a \in h_k = 0$ then $a \sim \mathcal{U}\{-1, 1\}$
7: $\quad H_k = \mathcal{F} h_k$
8: $\quad$ Extract $\theta_{sk(new)}$ from $|H_k| e^{j\theta_k}$
9: $\quad$ **if** $\theta_{sk(new)} \neq \theta_{sk(current)}$ **then**
10: $\qquad \theta_{sk(current)} \leftarrow \theta_{sk(new)}$
11: $\quad$ **else**
12: $\qquad$ Loop is set to false
13: $\quad$ **end if**
14: **until** Loop is set to false

To have a better visual comprehension of DIBS Figure 1 shows a conceptual instance. Figure 1.a is a 64 points DIBS. This signal is obtained in step 6 of Algorithm 1. Figure 1.b shows the 32 Fourier coefficients contained in H_k. These coefficients are used to evaluate P. Figure 1.c shows phase angles θ_k of the respective H_k Fourier coefficients. Terms with the subscript $_s$ in addition to $_k$ such as θ_{sk} refer to desired Fourier coefficients vector only, not the total spectrum.

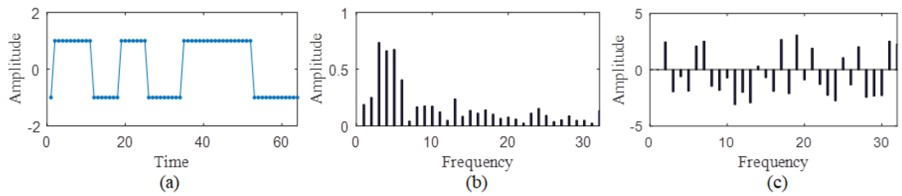

Fig. 1. Example of DIBS (a) DIBS in the time domain (b) Fourier coefficients $|H_k|$ in the frequency domain (c) phase angle θ_k of the respective Fourier coefficient in the frequency domain

This algorithm improves a single solution iteratively. The algorithm converges when the phase angle vector $\theta_{sk(new)}$ obtained at the end of each iteration is equal to $\theta_{sk(current)}$ used at the beginning of the iteration. At this moment Algorithm 1 is stagnated and successive iterations will give the same result $\theta_{sk(new)}$. This vector is the solution given by the algorithm. When the algorithm converges, P is minimized for that execution, but it does not guarantee an optimal solution [10]. We can run Algorithm 1 from step 4 to 6 to get the DIBS signal with the obtained solution $\theta_{sk(new)}$. This algorithm can be executed multiple times in order to try to find a better minimum of P.

The main problem of this method is the lack of strategies to improve the result. Because of this, PSO as a population-based method is proposed to find better solutions.

4 PSO for optimizing DIBS

PSO is a population based meta-heuristic algorithm. Populations are decentralized and composed by homogeneous particles. Each particle represents a potential solution and moves around the search space. Particles only have two informations, the historic best solution that is called cognitive component and the swarm best solution called social component. The algorithm works iteratively, updating the position of every particle of the population individually based on the cognitive and social component of the particle.

The standard continuous PSO is employed in this paper[17]. A basic pseudocode for this algorithm is described in Algorithm 2.

Algorithm 2 Canonical PSO

Initialize random $x_{i,d}$ and $v_{i,d}$ for every particle i in every dimension d
repeat
  for each particle i **do**
    if $f(x_i) < f(pb_i)$ **then** ▷ Update particle best position
      $pb_i = x_i$
    end if
    if $f(x_i) < f(gb)$ **then** ▷ Update global best position
      $gb = x_i$
    end if
  end for
  for each particle i **do** ▷ Update particle velocity and position
    for each dimension d **do**
      $v_{i,d} = v_{i,d} + C_1 * Rnd(0,1)[pb_{i,d} - x_{i,d}] + C_2 * Rnd(0,1)[gb_d - x_{i,d}]$
      $x_{i,d} = x_{i,d} + v_{i,d}$
    end for
  end for
  $it = it + 1$
until it>number of iterations

The objective function of the PSO is represented in Algorithm 3. This is essentially the same objective function from Algorithm 1. In Algorithm 1 the possible solutions is the phase angle vector of the desired Fourier coefficients θ_{sk}. In the case of PSO, each particle is a possible solution. The PSO particles will be the phase angle vector θ_{sk} of the desired Fourier coefficients. For example, from Figure 1, if we want to optimize DIBS to get the maximum power in the Fourier coefficients $|H_2|$, $|H_4|$ and $|H_6|$ and reduce the harmonic distortion, the particles will be codified as $(\theta_2, \theta_4, \theta_6)$.

Taking in count that particles are phase angle vectors, they are codified in the continuous domain. Each element of the particle can take values from 0 to 2π. The length of the particle is the number of Fourier coefficients we want to create the signal.

Algorithm 3 Objective function of PSO

$D_k = |D_k| e^{j\theta_k}$ ▷ Particle is θ_{sk} contained in θ_k
$d_k = \mathcal{F}^{-1} D_k$
$h_k = sng(d_k)$; If $a \in h_k = 0$ then $a \sim \mathcal{U}\{-1, 1\}$
$H_k = \mathcal{F} h_k$
Extract $\theta_{sk(new)}$ from $|H_k| e^{j\theta_k}$
$P = \sum_{k=1}^{n} |H_k| / \sum_{k=1}^{n} |H_{sk}|$ ▷ From equation 1. P is the fitness value

As we can identify from the objective function, the input variables are the phase angle vector θ_{sk}. The dimensionality of the input vector is the same as the number of Fourier coefficients we desire.

The objective function has parameters to be defined. This will define the characteristics of the DIBS we are looking for. These characteristics are the amplitude and and the position in the frequency spectrum of the desired Fourier coefficients. They take positions from 0 to $N/2$, being N the number of the DIBS points in the time domain. $|D_{sk}|$ represents the amplitude of the desired Fourier coefficients and it is contained in $|D_k|$.

5 Benchmark Instances and Experiment Setup

In order to validate the proposed solution for optimizing DIBS, 8 different sets of parameters for the objective function in Algorithm 3 are defined and shown in Table 1. Instances 1 to 5 are taken from the paper where Van Den Bos proposed his algorithm [10]; 6, 7 and 8 are added to complete different scenarios. Second column shows the quantity of the desired Fourier coefficients representing the dimensionality of the PSO particles. Third and fourth columns represent the position and the amplitude ratio of each desired Fourier coefficients respectively. Fifth column shows the number of points of the DIBS instance. Instances 1 to 5 show different particle dimensionality, different Fourier coefficient positions and different amplitude ratios. Instances 5 to 8 show different number of points for DIBS.

Table 1. Problem instances

	Quantity of desired harmonics	Harmonic position	Amplitude ratio	N
1	3	1,16,32	1,1,1	256
2	5	1,3,7,15,31	$1, \sqrt{2}, 1, \sqrt{2}, 1$	256
3	3	1,6,10	1,3,5	256
4	9	3,17,18,19,28,29,30,31,32	$1, \sqrt{3}, \sqrt{3}, \sqrt{3}, \sqrt{5}, \sqrt{5}, \sqrt{5}, \sqrt{5}, \sqrt{5}$	256
5	16	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	128
6	16	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	256
7	16	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	512
8	16	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16	1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1	1024

Population size is 50 particles. Cognitive and social components are C1 = C2 = 2. Stop criterion is 800 iterations giving an amount of 40,000 function evaluations. These were set empirically. The minimum and maximum speed are set to 0 and 2π respectively. No constriction or inertia coefficients were needed.

To have a reference of the quality of the solutions from PSO it is compared with Van Den Bos algorithm described in Algorithm 1. Because of the objective function for both algorithms is the same, the stop criterion of 40,000 function evaluations is also used for the Van Den Bos algorithm. Each time the Van Den Bos algorithm converges it will be executed again tracking the number of function evaluations until reaching the stop criterion. The best result will be updated every time after a better result is obtained.

The hardware used to run the algorithm is an Intel Core i7-3632QM and 8 GB RAM memory running Windows 7. Language employed for implementing the algorithm is C. Source code is available in Github[1].

Every experiment is executed 30 times by both algorithms. Average value and standard deviation are calculated. Percentage improvement is calculated in order to appreciate the difference between average results. At last, non-parametric signed rank Wilcoxon test is applied to the result of the 30 executions given by each algorithm for each instance. Result is given as two-tailed p-value. Signed rank Wilcoxon test does not give us any information about how much one solution is better than the other, it only guarantee us that the populations of the results from both algorithms are statistically different and not a product of the causality with a confidence level of 95%. Tests are performed in the statistical software R.

6 Results and Analysis

Table 2 shows the results for P including average deviation of both Van Den Boss and PSO algorithms, percentage of improvement, and the two-tailed Wilcoxon rank sum test p-values. Instances can be divided in two sets: from 1 to 4 and 6 have the same numbers of points but different Fourier coefficient quantity and amplitudes; from 5 to 8 the properties of the Fourier coefficients are the same but the number of points increases.

Table 2. Results obtained in experiments

	Van Den Bos algorithm	**PSO**	**Improvement (%)**	**p-value**
1	**3.9278**±0.000000	**3.9278**±0.000000	0	-
2	2.5391±0.000308	**2.4950**±0.008739	1.74	1.7344e-06
3	4.1390±0.000000	**4.1265**±0.006645	0.30	1.7344e-06
4	2.7197±0.000000	**2.6607**±0.013973	2.17	1.7344e-06
5	1.5460±0.001126	**1.4793**±0.004593	4.31	1.7344e-06
6	1.8040±0.003672	**1.7097**±0.009874	5.23	1.7344e-06
7	2.0690±0.000661	**1.9531**±0.011746	5.60	1.7344e-06
8	2.3112±0.001504	**2.1818**±0.012984	5.60	1.7344e-06

[1] Source code of Van den Bos and PSO-based algorithms for DIBS optimization: https://github.com/sfs325/DIBS-PSO-optimization

Instance 1 can be considered the one with less complexity having a small number of Fourier coefficients and equal amplitude of them. Results show the same performance for both algorithms. Instance 2 increases the complexity compared with instance 1 in the number of Fourier coefficients from 3 to 5 and different amplitudes for the coefficients. PSO application gets a better result by 1.74%. Instance 3 has a PSO improvement of 0.30%. Instance 4 has 9 Fourier coefficients with different amplitudes. In this case the improvement is 2.17%. Instance 6 has 16 desired Fourier coefficients and the biggest improvement of 5.23%.

Instances from 5 to 8 keep the same characteristics for the Fourier coefficients but the number of points increases from 128 to 1024. The lowest increase is found with 128 points being 4.31% and increases gradually with the number of points to 5.60% when 1024 points are used.

With these results some facts can be determined. Instances 2, 3, 4 and 6 with the same number of points of 256 show a proportional increment in the improvement based on the number of desired Fourier coefficients. Instances 5, 6, 7 and 8 with the same Fourier coefficient conditions, the improvement of PSO increases gradually with number of points. Standard deviations are small showing robustness for both algorithms.

Wilcoxon rank sum test p-value shows for the first instance that the results of both algorithms are statistically identical. However, for the remaining instances, p-values are really small and meets the premise of being less than 0.05 to be considered statistically different with a confidence level of 95%. It can be considered that the proposed solution statistically improves the optimization of DIBS.

When working with iterative algorithms it is important to monitor the convergence in order to analyze the behavior of the algorithm. Figure 2.a and 3.a shows convergence for Van Den Bos algorithm in orange and PSO in blue. Figure 2.b and 3.b shows diversity for PSO. In the first instance both algorithms converged to the same result always as seen in Table 2, even though, PSO converges faster. In instances from 2 to 8, Van Den Bos algorithm converges much faster than PSO. It only needs less than 2,000 function evaluations to get stagnated for every instance. On the other hand, PSO shows a constant improvement of the result. At the end of the 40000 evaluations it can not be considered as totally stagnated but close enough to give an acceptable solution.

The diversity for every instance seen in Figure 2.b Figure 3.b confirms a healthy convergence of the PSO population, not too fast nor too slow. This is important to detect problems such as premature convergence or stagnation in local minima [18].

Execution time had no significant changes in instances with the same number of points for both algorithms. This is because the most resource demanding operations are DFT and IDFT where the time for these operations depends on the number of points. The average time per execution for every experiment for both algorithms was 1.4 seconds for 128 points, 2.5 seconds for 256 points, 4.9 seconds for 512 and 10.1 seconds for 1024 points.

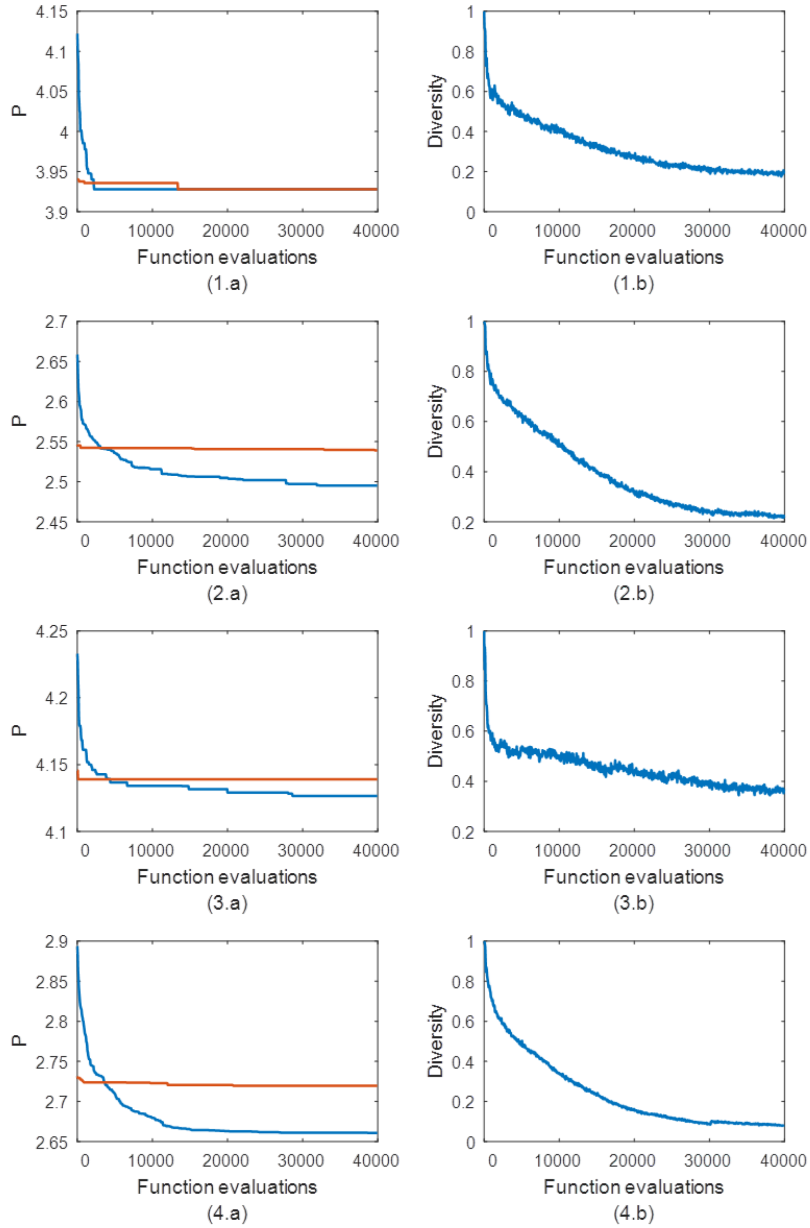

Fig. 2. Average values of Van den Bos and PSO algorithms executions for experiments from 1 to 4 (a) PSO convergence in blue, Van Den Bos algorithm convergence in brown, (b) PSO diversity

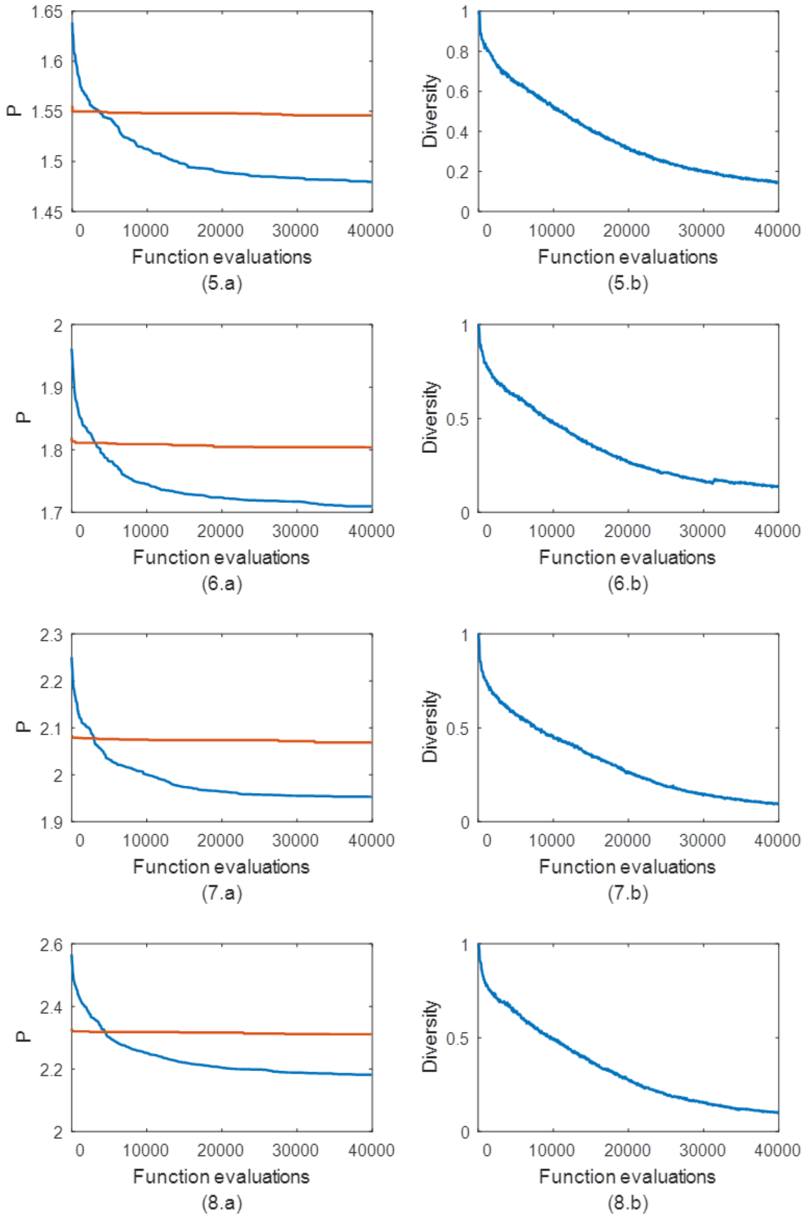

Fig. 3. Average values of Van den Bos and PSO algorithms executions for experiments from 5 to 8 (a) PSO convergence in blue, Van Den Bos algorithm convergence in brown, (b) PSO diversity

7 Conclusions and Future Work

From the problem of the lack of numerical methods for optimizing DIBS, alternative methods must be employed. The Van Den Bos algorithm has been widely used in the literature but it has a fast convergence to an aproximated value. It also lacks mechanism to control the convergence in order to improve the results. For this purpose PSO was applied to this problem in this work.

PSO has shown a good response solving this problem. Graphs show a proper convergence for every instance. Diversity looks very healthy so it is not needed to implement additional strategies to control it.

According to the percentage improvement in conjunction with p-value obtained from the Wilcoxon rank sum test, PSO statistically offers better DIBS optimization than Van Den Bos Algorithm. Apart from the first instance where the complexity was the least, PSO gave a better result than Van Den Bos algorithm for every execution. This shows how powerful are heuristics algorithms for optimizing this type of problems such as PSO in this case.

A 5.60 % of improvement in the best case would seem small, but talking about signals for system identification every improvement in the input signal will improve the output measurements[2, 4, 5]. This will also increase the weight of DIBS in the balance when compared with other type of signals.

It has been observed that parameters with impact in the optimization are the number of desired Foutier coefficients and the number of points. As higher the number of desired Foutier coefficients or the number of points the better the optimization of PSO over Van Den Bos algorithm is.

As future work we intend to apply PSO to optimize DIBS using real life measurements of the model to be identified as feedback to calculate the fitness. This will include the model in the objective function of PSO making the optimization of DIBS even more precise.

References

1. Pintelon, R., Schoukens, J.: System Identification: A Frequency Domain Approach. Wiley (2001)
2. Sanchez, B., Vandersteen, G., Bragos, R., Schoukens, J.: Basics of broadband impedance spectroscopy measurements using periodic excitations. Measurement Science and Technology **23**(10) (2012)
3. Kamat, D.K., Patil, V., Bangal, C.B., Patil, P.M.: Comparative Analysis of Multisine and Binary Excitation Signals for Human Body Analysis. **5**(5) (2015)
4. Rhee, H., Jang, K.S., Shin, M.J., Lee, J.W., Kim, I.Y., Song, S.H., Lee, D.W., Lee, S.B., Kwak, I.S., Seong, E.Y.: Use of multifrequency bioimpedance analysis in male patients with acute kidney injury who are undergoing continuous venovenous hemodiafiltration. PLoS ONE **10**(7) (2015)
5. Cornish, B.H., Thomas, B.J., Ward, L.C.: Improved prediction of extracellular and total body water using impedance loci generated by multiple frequency bioelectrical impedance analysis. Physics in Medicine and Biology **38**(3) (1993)
6. Williams, J.: The Art and Science of Analog Circuit Design. Butterworth-Heinemann (1998)

7. Godfrey, K.R., Tan, A.H., Barker, H.A., Chong, B.: A survey of readily accessible perturbation signals for system identification in the frequency domain. Control Engineering Practice **13**(11) (2005) 1391–1402
8. Ojarand, J., Min, M., Annus, P.: Crest factor optimization of the multisine waveform for bioimpedance spectroscopy. Physiological measurement **35**(6) (2014) 1019–33
9. Friese, M.: Multitone signals with low crest factor. IEEE Transactions on Communications **45**(10) (1997) 1338–1344
10. Taylor, P., Bos, A.V.A.N.D.E.N., Krol, R.G.: Synthesis of discrete-interval binary signals with specified Fourier amplitude spectra. International Journal of Control **30**(5) (2007) 871 – 884
11. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. International Journal of Bio-Inspired Computation **3**(1) (2011) 1–16
12. Badamchizadeh, M., Madani, K.: Applying Modified Discrete Particle Swarm Optimization Algorithm and Genetic Algorithm for system identification. Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on **5** (2010) 354–358
13. Chang, W.D.: Nonlinear system identification and control using a real-coded genetic algorithm. Applied Mathematical Modelling **31**(3) (2007) 541–550
14. Jeong, I.k., Lee, J.j.: Adaptive simulated annealing genetic algorithm for system identification. Engineering Applications of Artificial Intelligence **9**(5) (1996) 523–532
15. Horner, A.: A genetic algorithm-based method for synthesis of low peak amplitude signals. Journal of the Acoustical Society of **99**(1) (1996) 433–443
16. Niedostatkiewicz, M., Zielonko, R.: Investigation on accelerated impedance spectrum measurement method with multisine signal stimulation. Metrology and Measurement Systems **16**(4) (2009) 619–630
17. Kennedy, J., Eberhart, R.: Particle swarm optimization. Neural Networks, 1995. Proceedings., IEEE International Conference on **4** (1995) 1942–1948
18. Pant, M., Thangaraj, R., Abraham, A.: Particle Swarm Optimization : Performance Tuning and Empirical Analysis. Foundations **3** (2009) 101–128