applied sciences

MDPI

Article

An Enhanced Genetic Algorithm for Parameter Estimation of Sinusoidal Signals

Chao Jiang [1,2], Pruthvi Serrao [2], Mingjie Liu [1] and Chongdu Cho [3,*]

[1] School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; chiaojiang@inha.edu (C.J.); liumj@cqupt.edu.cn (M.L.)

[2] Department of Mechanical Engineering, Graduate School, Inha University, Incheon 22212, Korea; serrao@inha.edu

[3] Department of Mechanical Engineering, Inha University, Incheon 22212, Korea

* Correspondence: cdcho@inha.ac.kr; Tel.: +82-32-860-7321

Received: 17 June 2020; Accepted: 23 July 2020; Published: 25 July 2020

Abstract: Estimating the parameters of sinusoidal signals is a fundamental problem in signal processing and in time-series analysis. Although various genetic algorithms and their hybrids have been introduced to the field, the problems pertaining to complex implementation, premature convergence, and accuracy are still unsolved. To overcome these drawbacks, an enhanced genetic algorithm (EGA) based on biological evolutionary and mathematical ecological theory is originally proposed in this study; wherein a prejudice-free selection mechanism, a two-step crossover (TSC), and an adaptive mutation strategy are designed to preserve population diversity and to maintain a synergy between convergence and search ability. In order to validate the performance, benchmark function-based studies are conducted, and the results are compared with that of the standard genetic algorithm (SGA), the particle swarm optimization (PSO), the cuckoo search (CS), and the cloud model-based genetic algorithm (CMGA). The results reveal that the proposed method outperforms the others in terms of accuracy, convergence speed, and robustness against noise. Finally, parameter estimations of real-life sinusoidal signals are performed, validating the superiority and effectiveness of the proposed method.

Keywords: sinusoidal signal; parameter estimation; genetic algorithm (GA); benchmark function

1. Introduction

Many practical signals, such as voice and audio signals, power system transient signals, and response signals of sensors, are recognized as signals of the sum of sinusoidal signals. The parameter estimation of sinusoidal signals from the periodic time series data is a classical problem of ongoing interest in signal processing. In recent years, it has drawn considerable attention of scientists and researchers from various fields including speech analysis [1], electrical power and energy system [2], and sensor signal analysis [3], among others. There are some methods addressing this problem; in general, these can be classified into two categories, classical techniques [3–10], and intelligent algorithms [11–20]. The classical techniques, when compared with intelligent ones, are easier to implement and are computationally more efficient. However, they do have limitations, including frequency resolution, low accuracy, mathematical demands, and so on. On the other hand, the intelligent algorithms and hybrids are favorably employed due to their fewer requirements about the problems and excellent resolution. A few recent applications include atomic physics [21], electrical machines [22], big data [23], traveling salesman problem [24], and fuel cell [25]. Thus, intelligent techniques have extended significantly to provide effective and reliable solutions for parameter estimations of sinusoidal signals [11–20,26–29] accordingly.

In a literature [11], Spavieri et al., used a particle swarm optimization (PSO)-based approach for the parameterization of a power capacitor model fed by harmonic voltages in power distribution systems. For parameter estimation of exponentially damped sinusoidal signals, Xiao et al. [15] presented a specific artificial neural network (ANN) based estimator, by which the parameters of each exponentially damped sinusoidal signal component could be directly calculated with a high precision. However, the performance strongly relies on the converged weights of the ANN, and hence, an absence of optimal weights or insufficient training will result in a poor performance. Based on a genetic algorithm (GA), Coury et al. [18,20] presented an efficient estimator to evaluate the frequency and phase for a phasor management unit. In comparison to the phase-locked loop (PLL) and the discrete Fourier transform (DFT) based methods, the GA revealed the fastest response to a sufficient precision. In order to minimize bias and improve estimation performance, researchers generally combine different methods to overcome the individual drawbacks. In another article, Djurović et al. [17] presented a hybrid genetic algorithm (HGA) fused with a maximum likelihood approach to estimate the phase parameters of frequency modulated signals. Zang et al. [30] proposed a cloud model-based genetic algorithm (CMGA) for optimization problems. Benefiting from the cloud model, the CMGA continuously produces new individuals, and hence, the exploration and randomness can be significantly improved, showing a better optimization when compared to other methods. In the literatures [31–33], Raja et al., introduced some hybrid techniques with ANN, GA, and other algorithms for the dynamics of nonlinear singular heat conduction model of the human head, nonlinear Painlevé II systems, and nonlinear singular Thomas–Fermi systems. Comparisons between the proposed schemes and the standard numerical solutions as well as analytical methods revealed the feasibility and effectiveness of the proposed schemes. Similarly, a hybrid approach using a differential evolution (DE) and GA was proposed by Ali et al. [34], where a DE-based multi-parent crossover operation was introduced to enhance the search ability and to avoid premature convergence by exploring more solutions in the problem search space. In combination with a local search optimization algorithms, Zaman et al. [35] proposed an HGA to estimate the amplitude, frequency, range, elevation angle, and azimuth angle of near field sources; the proposed schemes achieved good results in terms of accuracy, convergence rate, and robustness, but the complexity increased a lot.

In the light of these facts, authors are motivated in this work to design an enhanced genetic algorithm (EGA), through a prejudice-free selection mechanism for preserving population diversity, diversified individuals with a two-step crossover (TSC) operator, and a good cooperation of fast convergence and searching performance with an adaptive mutation strategy, for estimating the parameters of sinusoidal signals. The prominent features of the proposed method are briefly summarized as follows:

- Validation through the results of a comparative analysis in terms of performance-monitoring metrics based on the mean, coefficients of determination (R^2), and the sum of squared residuals value (SSR) of each algorithm;
- An intelligible concept with easy realization and a worry-free tradeoff between global and local search in comparison to general hybrid techniques.

The rest of the work is organized as follows. The mathematical ecological theory foundation of the proposed approach is presented in Section 2. Section 3 describes the proposed EGA in detail, including the prejudice-free selection, the two-step crossover operator, and the adaptive mutation operator. In Section 4, benchmark function based numerical analyses are carried out to verify the validity of the EGA. The parameter estimations of real-life sinusoidal signals are performed and discussed in Section 5. At last, we summarize our work and highlight the main contributions in Section 6.

2. Mathematical Ecological Theory Foundation

In ecology, the number of offspring restricted to be equal to or less than that of parents is not common. Even though this may happen for reasons such as disease, food, water, and other factors, the species may become extinct or be restricted due to genetic competition as well. When the number of offspring is more than two, the population size grows rapidly with more competitive individuals [24], resulting in an increased chance for producing better offspring. Thus, it is desirable to have a larger number of individuals in genetic algorithms.

Population dynamics were first studied by Verhulst [36] in 1838, who describes the size and age composition of populations as dynamical systems driven by biological and environmental processes. The population dynamics model of a species can be mathematically expressed by:

$$\frac{dP}{dt} = rP \cdot (1 - \frac{P}{K}), \tag{1}$$

where P represents population size, t is time, r defines the growth rate, and K is a constant denoting the environment carrying capacity. The solution to Equation (1) with an initial population P_0 can be written as:

$$P(t) = \frac{KP_0 \cdot e^{rt}}{K + P_0 \cdot (e^{rt} - 1)}. \tag{2}$$

Taking the birth and mortality rate into account, Equation (2) can be rewritten as:

$$\Delta P(t) = \frac{KP_0 \cdot e^{(r_1 - r_2)t}}{K + P_0 \cdot [e^{(r_1 - r_2)t} - 1]}, \tag{3}$$

where ΔP is the total population size, and r_1 and r_2 are birth rate and mortality rate, respectively. Over time, the population dynamics presents different consequences in accordance with the relations between r_1 and r_2.

1. When the birth rate is higher than the mortality rate, $r_1 > r_2$, the final population size for an infinite amount of time, t, can be simplified as:

$$\lim_{t \to \infty} \Delta P(t) = \lim_{t \to \infty} \frac{KP_0 \cdot e^{(r_1 - r_2)t}}{K + P_0 \cdot [e^{(r_1 - r_2)t} - 1]} = K \tag{4}$$

   This means that a species can exist forever.

2. When the birth rate is equal to the mortality rate, $r_1 = r_2$, the exponential terms in the numerator and denominator become 1, so Equation (3) is given as follows:

$$\lim_{t \to \infty} \Delta P(t) = P_0. \tag{5}$$

   Here, as time approaches infinity, the final population size will equal to the initial value. What if the initial population size equals to 1 or 2? Although this species can live for a long time, the problem of self-reproduction and inbreeding results in an extinction.

3. When the birth rate is smaller than the mortality rate, $r_1 < r_2$, it is clear that $e^{-x} \to 0$ as $t \to \infty$. Hence, one can get the following expression:

$$\lim_{t \to \infty} \Delta P(t) = \lim_{t \to \infty} \frac{KP_0 \cdot e^{-(r_2 - r_1)t}}{K + P_0 \cdot [e^{-(r_2 - r_1)t} - 1]} = 0 \tag{6}$$

   Obviously, this equation represents an inevitable extinction of a species.

According to the analysis above, on the one hand, it can be concluded that the extinction of a species occurs if the birth rate is equal to or less than the mortality rate; only when $r_1 > r_2$ can the species survive forever. On the other hand, it is found that the final population size strongly depends on both the growth rate and the time when the initial population size is fixed. Thus, in order to increase the number of offspring and get excellent individuals within a limited time, it is very necessary to improve the reproduction rate of a species [24]. In the case of the standard genetic algorithm (SGA), as the procreation is artificially controlled to be unchangeable, the extinction of the population of SGA is not allowed. Hence, neither is the evolutionary law consistent with the natural law, nor will it have a chance to acquire more excellent individuals. In this paper, based on the biology theory and mathematical ecological theory foundation, a genetic algorithm with several enhancements is proposed in order to achieve a higher accuracy, and faster convergence, and hence, to further improve the parameter estimation results of sinusoidal signals.

3. The Proposed EGA

3.1. Prejudice-Free Selection

Elitism-based simple selection strategies [37] are popularly used in GAs, which favor the higher fitness chromosomes (individuals) and discriminate against the lower ones. Consequently, outstanding individuals dominate in iterations leading to a monotonous population diversity and breaking up a healthy competition. Besides, in the SGA the production mechanism of offspring using a random pairing through two individuals will reduce the population quality, because an outstanding chromosome may combine with a poor one and lose its superior genes. To settle this, in the EGA the individuals are divided into two groups according to their fitness values: the best half individuals with high fitness values are in a benign group, while the remaining ones are assigned to a malignant group; the evolution in each group takes place independently, and a new selection is always executed to redistribute them in every iteration. So, offspring may stay in their original group or migrate to the other in accordance with their new fitness values. Compared to traditional selections, this operation not only reduces the overhead of selection resulting from a non-bias selection procedure [38], but also it allows a parallel computation providing a faster convergence [39]. Thanks to this, the population diversity and efficient convergence can be guaranteed.

3.2. Two-Step Crossover (TSC) Operator

In GAs, crossover operation simulates the processes of sexual reproduction, through which offspring inherit parts of genetic information from their parents. However, conventional crossover operators just follow the genetic concept, weakening the search ability [40,41]. For instance, in a single-point crossover (SPC), two individuals are first selected as the parent chromosomes, then a randomly-generated point is used to truncate the chromosomes and get two sub-sequences; the fragments behind the point are exchanged with each other and coupled with the front ones to generate two new offspring. The principle of a SPC operation is shown in Figure 1.

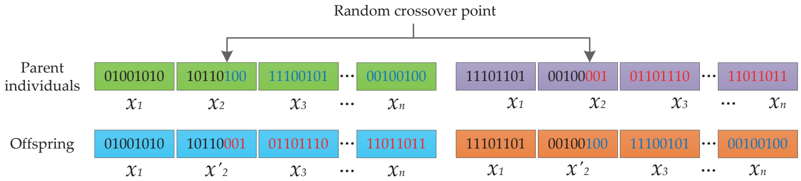


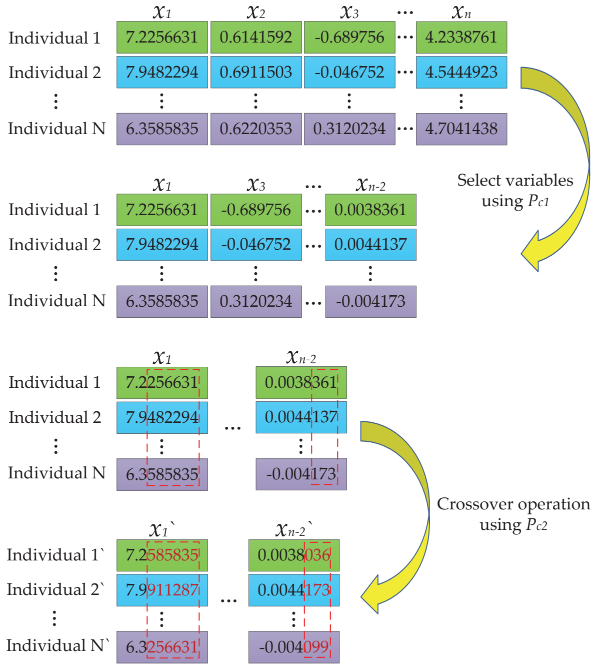
Figure 1. Single-point crossover operation for binary sequence.

For a given population of a size of $N \times M$, the number of the newly generated variables after a SPC operation can be calculated by $[NP_{c2}]$, where N is the number of individuals, M denotes the length of each chromosome, $M = n \times l$, n is the number of the variables defined by the chromosomes, l is the coding length of each variable, and P_{c2} is the crossover probability. The symbol $[\cdot]$ rounds the number toward positive infinity. It is clear that a SPC can produce new individuals and change the population diversity. However, only the variables whose sequences are trimmed are updated during the operation, and hence, the population diversity is still restricted. Although other types of crossover operations such as two-point crossover and multi-point crossover can mitigate this effect, the similar operation mechanism dooms the same outcome.

In this regard, a two-step crossover strategy that comprehensively considers the information exchange among entities and variables is designed in this work. The details are described as follows:

1. Dividing each chromosome into segments based on the number of the variables to be solved, and gathering all the specific segments for a certain variable to form a variable set for the related variable;
2. Randomly selecting some variable sets with a probability P_{c1};
3. Implementing crossover operation on each elected set to update the variables with a probability P_{c2};
4. Once the crossover is completed, reassembling the variable sets together to get the new individuals.

For a better understanding, a SPC is employed in the TSC to illustrate the mechanism, as shown in Figure 2. Here, the number of the newly produced variables by the TSC can be calculated by $[nP_{c1}] \times [NP_{c2}]$, where P_{c1} is the first crossover probability, and the other parameters are the same as the ones mentioned above. The difference between the newly generated variables between the TSC and SPC is mathematically expressed as $[nP_{c1}] \times [NP_{c2}] - [NP_{c2}]$. With some mathematical manipulations, this expression can be simplified as $[nNP_{c2}] \cdot (P_{c1} - 1/n)$. It can be known that a positive value is always achieved if $(P_{c1} - 1/n) > 0$. So, by simply adjusting P_{c1}, the number of the updated variables yielded by the TSC can be much more than that of the SPC, which amounts to indirectly increase the number of the offspring [24]. Hence, the population diversity can be significantly improved. Besides, notice that the applied SPC in the TSC can also be replaced by other types of crossover operations; for the case of multi-point crossover, a considerable amount of new variables can be obtained.



	x_1	x_2	x_3	⋯	x_n
Individual 1	7.2256631	0.6141592	-0.689756	⋯	4.2338761
Individual 2	7.9482294	0.6911503	-0.046752	⋯	4.5444923
⋮	⋮	⋮	⋮		⋮
Individual N	6.3585835	0.6220353	0.3120234	⋯	4.7041438

	x_1	x_3	⋯	x_{n-2}
Individual 1	7.2256631	-0.689756	⋯	0.0038361
Individual 2	7.9482294	-0.046752	⋯	0.0044137
⋮	⋮	⋮		⋮
Individual N	6.3585835	0.3120234	⋯	-0.004173

	x_1	⋯	x_{n-2}
Individual 1	7.2256631		0.0038361
Individual 2	7.9482294	⋯	0.0044137
⋮	⋮		⋮
Individual N	6.3585835		-0.004173

	x_1`	⋯	x_{n-2}`
Individual 1`	7.2585835		0.0038036
Individual 2`	7.9911287	⋯	0.0044173
⋮	⋮		⋮
Individual N`	6.3256631		-0.004099

Figure 2. The two-step crossover operation for floating sequence.

3.3. Adaptive Mutation Operator

The mutation operator has a strong influence on population diversity and convergence speed, playing a significant role in GAs. Usually, mutation probability, P_m, keeps constant throughout the whole iteration in the SGA and HGAs [16,19,36,38,41]. Although a constant mutation probability provides GAs a relatively stable performance, the problem of ever-increasing similar individuals cannot be effectively solved. To address this, an adaptive mutation mechanism that considers the evolutionary features of GAs at different stages is proposed in this work.

At the beginning of an evolution, a small P_m is allocated for boosting the convergence [42]. As the processing progresses, the number of similar individuals increases rapidly, which causes the optimization to fall back to local optima repeatedly. In order to overcome this, a larger mutation probability is assigned at the intermediate stage so that the evolution can easily get rid of predicaments. In the final stage, a small probability is distributed again to ensure a global convergence [37].

Inspired also by the characteristics of population dynamics, a combination of a general Logistic function [36] and its mirror is employed to support the proposed idea in its entirety. The function is expressed as follows:

$$P_m = \begin{cases} a + \frac{b}{1+e^{-c(g-g_0)}}, g \leq G \\ a + \frac{b}{1+e^{+c(g-g_0)}}, g > G \end{cases} \tag{7}$$

where a denotes the initial mutation probability, b is the maximum value of the original Logistic curve, e is the natural logarithm base, c describes the steepness of the curve, g and g_0 are the evolution generation and the curve's midpoint, respectively. G is the total number of evolutionary iterations. Figure 3 shows the curves of the functions.

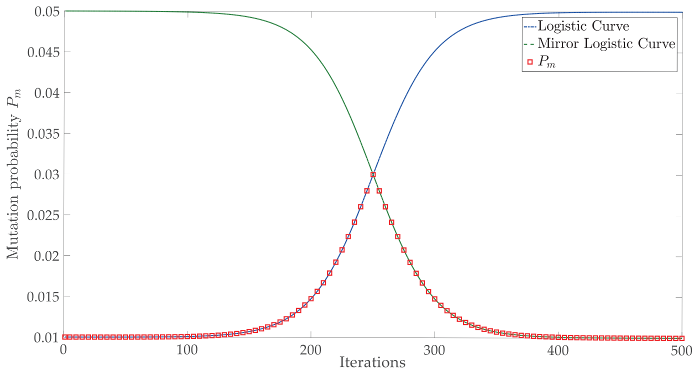


Figure 3. Adaptive mutation probability as the change of iteration.

3.4. EGA Procedures

To solve a problem using the proposed EGA, N individuals are randomly generated by a float encoding method as the initial population $Init\_Pop$. Each individual's objective value and fitness value are calculated, and the best individual $BestX\_Old$ in the current population is obtained. In order to keep population diversity and promote a healthy competition, individuals are assigned to a *Benign* group or *Malignant* group with the same size of $N/2$, respectively, according to their fitness values. As the individuals in the *Benign* group have higher fitness values, performing a crossover operation in this group is helpful to generate better descendants. Therefore, the TSC operation with probabilities P_{c1} and P_{c2} is run for them only. Instead of being discarded directly, in the *Malignant* group, the remaining individuals with a low fitness but rich population diversity go through the adaptive mutation. Once the crossover and mutation are complete, the fitness values of the individuals in the new population New_Pop will be evaluated. The individuals in the *Malignant* group that have a high fitness will migrate to the *Benign* group, and a reverse movement is also carried out synchronously for the ones in the *Benign* group having a low fitness. Again, the best individual $Best\_New$ in the current population will be picked up against the $Best\_Old$, and the better one is saved as $Best\_Temp$. As the optimum may hide near $Best\_Temp$, some random individuals (e.g., 10) are generated in this area, and their fitness values are calculated. If someone is better than $Best\_Temp$, they will be reassigned as $Best\_Temp$; else, $Best\_Temp$ keeps its original value. At the end of each iteration, $BestX\_Old$ and Old_Pop are updated by $Best\_Temp$ and New_Pop, respectively.

The termination criterion will be triggered when the maximum iteration $MaxGen$ reaches. A more clear procedure of the EGA is shown in Figure 4 and Algorithm 1.

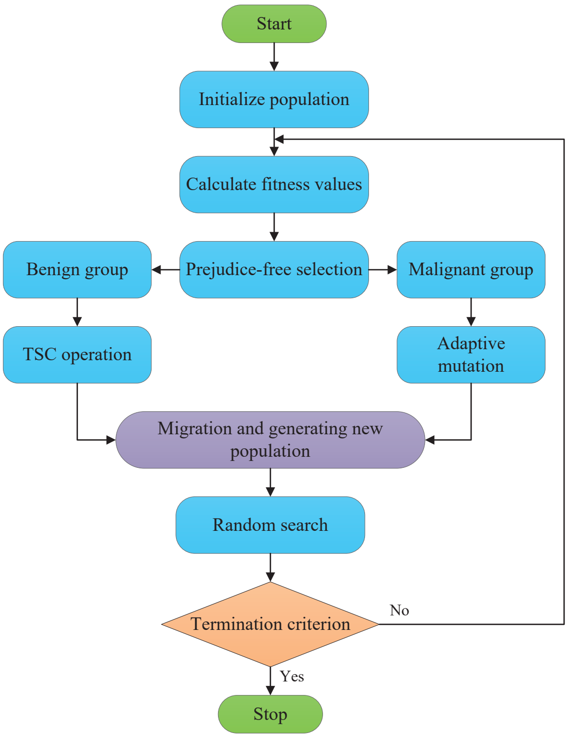


Figure 4. Flow chart of the proposed enhanced genetic algorithm (EGA).

Algorithm 1 The pseudo-code of the enhanced genetic algorithm.

```
Input: $N$ : size of initial population
  $MaxGen$ : maximum number of iteration
  $[P_{c1}, P_{c2}]$ : probabilities of the two-step crossover operation
  $[P_{m1}, P_{m2}]$ : probabilities of the initial and final adaptive mutation operation
  $num$ : number for generating new individuals around the best solution in each
        iteration
  $f(x)$ : objective function
Output: $Trace$ : the best solution and corresponding objective value in each iteration
  function $EGA(N, MaxGen, [P_{c1}, P_{c2}], [P_{m1}, P_{m2}], num, f(x))$
    $Init\_Pop \leftarrow random\ individuals\ with\ a\ size\ of\ N;$
    $Old\_Pop \leftarrow Init\_Pop;$
    for $i \leq N$ do
      $Calculate\ the\ fitness\ value\ of\ each\ individual\ in\ Old\_Pop;$
    end for
    $gen \leftarrow 1;$
    while $gen \leq MaxGen$ do
      $Best\_Old \leftarrow the\ best\ individual\ in\ Old\_Pop;$
      Selection:
        $Benign \leftarrow N/2\ high\ fit\ individuals\ in\ Old\_Pop;$
        $Malignant \leftarrow N/2\ low\ fit\ individuals\ in\ Old\_Pop;$
      Crossover:
        $Carry\ out\ the\ TSC\ operation\ to\ the\ individuals\ in\ the\ Benign\ group;$
```

```
            Mutation:
                $Apply\ an\ adaptive\ mutation\ operation\ to\ the\ individuals\ in\ the\ Malignant\ group;$
            $New\_Pop \leftarrow combination\ of\ the\ processed\ individuals\ from\ the\ Benign\ and\ Malignant\ groups;$
            for $i \leq N$ do
                $Calculate\ the\ fitness\ value\ of\ each\ individual\ in\ New\_Pop;$
            end for
            $BestX\_New \leftarrow the\ best\ individual\ in\ New\_Pop;$
            $BestX\_Temp \leftarrow (BestX\_New < BestX\_Old)\ ?\ BestX\_New : BestX\_Old;$
            $BestX\_Temp \leftarrow (\mathbf{min}(\mathbf{rand}(num)) < BestX\_Temp)\ ?\ \mathbf{min}(\mathbf{rand}(num)) : BestX\_Temp;$
            $BestX\_Old \leftarrow BestX\_Temp;$
            $Old\_Pop \leftarrow New\_Pop;$
            $Trace \leftarrow the\ best\ solution\ and\ corresponding\ objective\ value\ in\ current\ generation;$
            $gen \leftarrow gen + 1;$
        end while
        Return $Trace;$
    end function
```

4. Benchmark Function Study

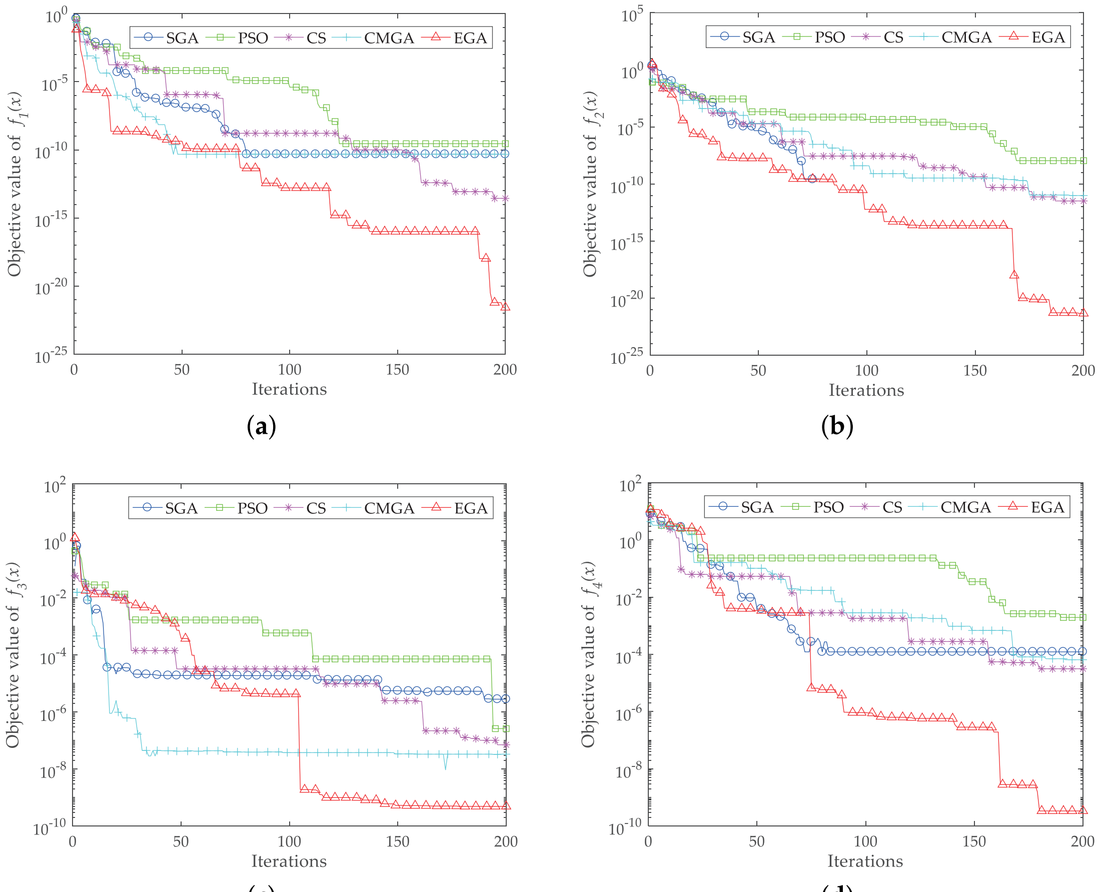
Benchmark functions [14,30,34,39,43] are widely adopted to demonstrate the performance of algorithms. Thus, in the present work, benchmark based numerical experiments are performed for the EGA as well. Here, eight unimodal and multimodal functions [40] are selected (see Appendix A).

As a comparison, studies using the SGA [44] as well as the PSO [11,29], the CS [28,29], and the CMGA [30] are also performed for the functions; the corresponding parameter configurations are listed in Table 1. Without favoritism, all the algorithms are assigned the same size initial population, $N = 50$, and the maximum iteration number, $MaxGen = 200$. The optimization results are plotted in Figures 5 and 6. To have an insight into how well these algorithms performed, all the vertical coordinates are displayed on a logarithmic scale.

Table 1. Parameter configurations of different algorithms for benchmark function optimization.

SGA	PSO	CS	CMGA	EGA
$N = 50$	$N = 50$	$N = 50$	$N = 50$	$N = 50$
$MaxGen = 200$	$MaxGen = 200$	$MaxGen = 200$	$MaxGen = 200$	$MaxGen = 200$
$P_c = 0.7$	$C_1 = 2, C_2 = 2$	$Pa = 0.25$	$m_1 \cdot t_1 = 1/600, n_1 = 0.1$	$[P_{c1}, P_{c2}] = [0.6, 0.7]$
$P_m = 0.01$	$w_1 = 0.9, w_2 = 0.4$	–	$m_2 \cdot t_2 = 1/600, n_2 = 0.1$	$[P_{m1}, P_{m2}] = [0.01, 0.1]$
–	–	–	$m_3 = 1/3, n_3 = 0.1$	–

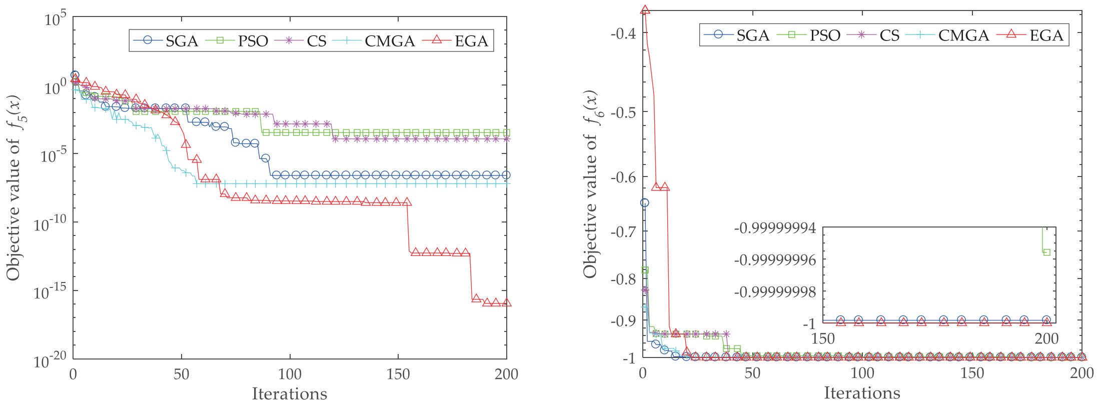
From Figures 5 and 6, it can be seen directly that the performance of the EGA is superior to that of the SGA, PSO, CS, and CMGA for most of the benchmark functions, except for the Zakharov function ($f_2(x)$). Examining the optimization results of $f_2(x)$ shown in Figure 5b, it is noted that the evolutionary curve of the SGA displays in part. This is caused by the estimating value of zero (equaling to the optimum), and the logarithmic y-axis. In this regard, the SGA performs better than the EGA. However, when compared to other results, the proposed EGA shows overwhelming advantages against others, especially for Yang's No. 5 function ($f_8(x)$). Different from other functions, $f_8(x)$ contains uniform distribution terms, which increases not only the number of local optima but also the uncertainty, and hence, the optimization becomes much more difficult. Taking a look at the results in Figure 6d, frequent vibrations are observed in the evolution curves of the SGA and PSO, while the EGA shows an unremitting evolution trend, demonstrating its robustness against noise.



(**a**) (**b**)

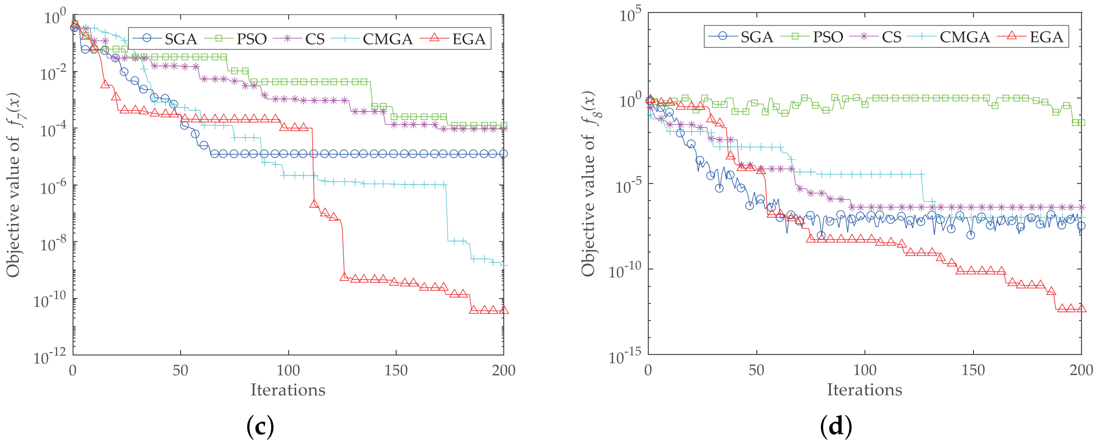
(**c**) (**d**)

Figure 5. Evolutionary curves of unimodal benchmark functions. (**a**) Sphere function. (**b**) Zakharov function. (**c**) Rosenbrock function. (**d**) Ackley function.



(**a**) (**b**)

Figure 6. *Cont.*



(**c**) (**d**)

Figure 6. Evolutionary curves of multimodal benchmark functions. (**a**) Griewank function. (**b**) WaveDrop function. (**c**) Xin-she Yang's No.2 function. (**d**) Xin-she Yang's No.5 function.

To further examine the performance, the obtained best and mean solutions as well as the convergence speed of each function were compared and listed in Tables 2–4, respectively. All the results were obtained over 50 Monte-Carlo simulations. Through Table 2, it is evident that the EGA outperforms the SGA, PSO, CS, and CMGA for most cases. Looking at the optimization results for $f_2(x)$, although only the SGA achieves the global optimum, the EGA still prevails over the others when comparing the rest optimizations. The results listed in Table 3 also confirm this point. In terms of the convergence speed, the maximum, the minimum, and the average termination iterations were measured, where the termination criterion was defined by $|O_{opt} - O_{obt}| < 10^{-4}$, O_{opt} is the optimum of a given problem, and O_{obt} is the obtained optimized result, respectively. From Table 4, one can find that the convergence speed of the proposed EGA is fast, presenting a strong competitiveness over its counterparts. From these comparisons, it can be concluded that the EGA shows an overwhelming superiority over the competitors in terms of accuracy, convergence speed, and robustness against noise.

Table 2. Comparison of the best optimal solutions of standard genetic algorithm (SGA), particle swarm optimization (PSO), cuckoo search (CS), cloud model-based genetic algorithm (CMGA), and EGA.

Function	Optimum	SGA	PSO	CS	CMGA	EGA
		F_{best}	F_{best}	F_{best}	F_{best}	F_{best}
$f_1(x)$	0	4.7684×10^{-11}	2.8466×10^{-10}	2.7727×10^{-14}	4.7684×10^{-11}	$\mathbf{2.7305 \times 10^{-22}}$
$f_2(x)$	0	**0**	1.0950×10^{-8}	3.2559×10^{-12}	9.7157×10^{-12}	4.5721×10^{-22}
$f_3(x)$	0	2.8240×10^{-6}	2.6240×10^{-7}	7.1331×10^{-8}	5.4599×10^{-9}	$\mathbf{4.9467 \times 10^{-10}}$
$f_4(x)$	0	1.2212×10^{-4}	0.0020	3.1397×10^{-5}	6.4976×10^{-5}	$\mathbf{3.3660 \times 10^{-10}}$
$f_5(x)$	0	2.4573×10^{-7}	0.0027	1.1657×10^{-4}	2.4573×10^{-7}	$\mathbf{1.1102 \times 10^{-16}}$
$f_6(x)$	-1	-1 [a]	-1 [a]	-1 [a]	-1 [a]	-1 [b]
$f_7(x)$	0	1.1984×10^{-5}	1.2576×10^{-4}	9.3775×10^{-5}	1.4512×10^{-9}	$\mathbf{3.6105 \times 10^{-11}}$
$f_8(x)$	0	3.1936×10^{-8}	0.0363	4.1578×10^{-7}	1.0680×10^{-7}	$\mathbf{4.5682 \times 10^{-13}}$

[a] the practical optimized value is very close to -1. [b] the practical optimized value is -1.

Table 3. Comparison of the average optimal solutions of SGA, PSO, CS, CMGA, and EGA.

Function	Optimum	SGA	PSO	CS	CMGA	EGA
		F_{mean}	F_{mean}	F_{mean}	F_{mean}	F_{mean}
$f_1(x)$	0	4.7684×10^{-11}	2.4852×10^{-6}	6.6205×10^{-7}	1.8619×10^{-9}	$\mathbf{2.4070 \times 10^{-13}}$
$f_2(x)$	0	**0**	1.1387×10^{-5}	2.3998×10^{-8}	2.3486×10^{-10}	9.8656×10^{-10}
$f_3(x)$	0	0.0625	7.9783×10^{-5}	1.6277×10^{-5}	$\mathbf{3.3094 \times 10^{-7}}$	1.1525×10^{-5}
$f_4(x)$	0	1.2212×10^{-4}	0.0331	4.2460×10^{-5}	7.5088×10^{-5}	$\mathbf{1.0119 \times 10^{-8}}$
$f_5(x)$	0	0.0539	0.0239	0.0113	1.9530×10^{-6}	$\mathbf{0.9103 \times 10^{-6}}$
$f_6(x)$	−1	−0.9319	−0.9719	−0.9864	−0.9955	**−0.9998**
$f_7(x)$	0	0.2374	0.0330	0.0431	6.0128×10^{-5}	$\mathbf{9.1184 \times 10^{-8}}$
$f_8(x)$	0	0.6746	0.5803	0.6875	1.6018×10^{-4}	$\mathbf{1.2212 \times 10^{-5}}$

Table 4. Comparison of the convergence speeds of SGA, PSO, CS, CMGA, and EGA.

Function	SGA		PSO		CS		CMGA		EGA	
	$G_{max(min)}$	G_{mean}	$G_{max(min)}$	G_{mean}	$G_{max(min)}$	G_{mean}	$G_{max(min)}$	G_{mean}	$G_{max(min)}$	G_{mean}
$f_1(x)$	45(7)	22.94	136(26)	64.10	41(10)	34.66	38(6)	28.16	**36**(**5**)	**18.28**
$f_2(x)$	54(**8**)	26.40	190(9)	101.86	65(27)	37.84	**40**(23)	35.26	44(12)	**25.84**
$f_3(x)$	200(13)	189.88	200(108)	135.40	187(30)	148.54	159(10)	121.63	**139**(**7**)	**108.44**
$f_4(x)$	200(200)	200.00	200(200)	200.00	200(120)	141.40	200(138)	120.33	**156**(**51**)	**100.72**
$f_5(x)$	200(200)	188.34	200(200)	200.00	200(120)	167.75	200(**30**)	**69.93**	200(51)	182.08
$f_6(x)$	200(48)	180.98	200(200)	189.58	200(108)	165.04	181(44)	140.71	**169**(**6**)	**124.98**
$f_7(x)$	200(39)	160.36	200(200)	200.00	200(105)	178.25	161(**37**)	153.24	**153**(39)	**119.04**
$f_8(x)$	200(19)	161.82	200(200)	200.00	200(42)	124.76	200(62)	**133.08**	200(**18**)	140.10

5. Parameter Estimation of Sinusoidal Signals

According to the numerical analyses above, the optimization performance of the EGA has already been demonstrated. To evaluate its performance for parameter estimations of sinusoidal signals, two real-life cases are selected, the voice data from singing the vowel 'ooh' [45], and the Circadian Rhythms [46]. As with the numerical analyses, the SGA, PSO, CS, and CMGA were also employed.

5.1. The Voice Dataset

The voice dataset gives the magnitudes of the voice when the vowel 'ooh' was sung at a pitch of 290 Hz. The frequencies and amplitudes found in the signal are used to determine the phonetic vowel, and are of interest in voice synthesis, therapy, and training. In the present work, a ten-parameter model is utilized for the signal as expressed as follows:

$$y(t) = K + \sum_{i=1}^{3} A_i \sin(\omega_i t + \varphi_i), \tag{8}$$

where K is the offset, A denotes the amplitude of sinusoid signal, ω and φ are the frequency and phase, respectively. The parameter initialization is set as $K \in [-1, 1]$, $A_i \in [0, 100]$, $\omega_i \in [-\pi/2, \pi/2]$, and $\varphi_i \in [-\pi/2, \pi/2]$.

The corresponding parameter configurations for the algorithms were the same as the ones in Table 1, except for the crossover and mutation parameters of the EGA. The new values were set as $[P_{c1}, P_{c2}] = [0.8, 0.7]$, and $[P_{m1}, P_{m2}] = [0.05, 0.22]$, respectively. The same $N = 200$, and $MaxGen = 500$ were assigned for all the algorithms as well.

Figure 7 shows the original data as well as the fitting curves, from which one can observe that all the curves follow the pattern well. For having an insight into the difference of the algorithms, the estimated parameters, and the coefficients of determination $\left(R^2\right)$ were examined, listed in Table 5. As can be seen from the table, the estimated parameters are very close, it is difficult to arrive at a conclusion. However, by comparing the values of the R^2, the performance of the EGA can be determined, which is the best among the algorithms. In other literature, Smyth et al. [47] considered the same dataset for frequency estimation, their reported result was 0.2299, 0.3408, and 0.1134, which are very close to the results in this work.

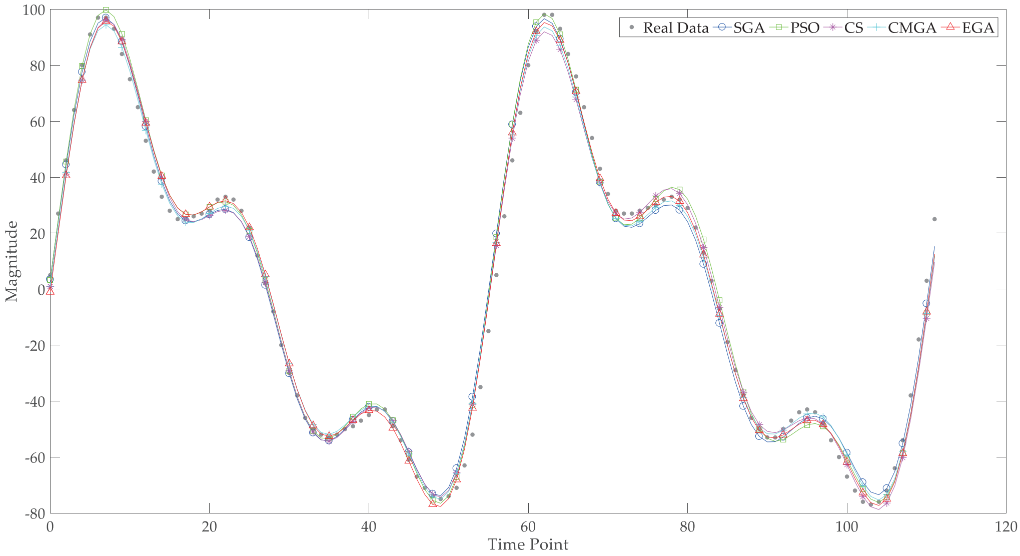


Figure 7. Plots of voice data and fitting curves.

Table 5. Estimation results of SGA, PSO, CS, CMGA, and EGA for voice dataset.

Parameter		SGA	PSO	CS	CMGA	EGA
Offset	K	−0.8437	0.1416	−0.9918	−0.9774	−0.9423
Amplitude	A_1	22.1220	22.7440	20.9413	22.1339	22.5130
	A_2	25.2844	26.5268	24.8365	24.9434	25.0975
	A_3	65.8691	66.8565	65.3737	64.6844	66.8589
Frequency	ω_1	0.2299	0.2320	0.2315	0.2293	0.2300
	ω_2	0.3404	0.3395	0.3418	0.3407	0.3407
	ω_3	0.1134	0.1126	0.1120	0.1133	0.1135
Phase	φ_1	−0.0569	−0.1467	−0.1465	−0.0216	−0.0609
	φ_2	−0.2428	−0.2082	−0.3285	−0.2656	−0.2611
	φ_3	0.1771	0.1797	0.2008	0.1409	0.1135
R^2		0.9909	0.9912	0.9916	0.9922	**0.9931**

5.2. The Circadian Rhythms

The Circadian Rhythms was an experiment that recorded the temperature of a long-tail pocket mouse every two minutes over three months. As some problems arose during the experiment, a proportion of outliers remained in the dataset. Though these outliers make the estimation quite difficult, they are meaningful and necessary for verifying the performance of an algorithm, because noise is ubiquitous in real-life signals. In this work, a 20 min temperature sample averaged from the dataset was extracted, and a four-component sinusoid model was used for the estimation, expressed as Equation (9):

$$y(t) = K + A\sin(\omega t + \varphi), \tag{9}$$

where K is the offset, A denotes the amplitude of sinusoid signal, ω and φ are the frequency and phase, respectively.

The parameter initialization for the estimation was set as $K \in [y_m - 50, y_m + 50]$, $A \in [-100, 100]$, $\omega \in [0, \pi]$, and $\varphi \in [0, \pi]$. Parameter y_m is the median of the real data. The parameter configurations for the algorithms were the same as those set in Section 5.1, except the ones for the EGA, whose new settings were: $[P_{c1}, P_{c2}] = [0.7, 0.7]$, and $[P_{m1}, P_{m2}] = [0.15, 0.35]$. The corresponding evaluated results are illustrated in Figure 8 and Table 6.

Figure 8 displays the original data and the best fitting curve. From this graph, it is evident to see that the dataset contains a large number of outliers, but the fitting curve successfully ignores them and follows nicely the periodic trend. In addition, the less evident outliers close to the fitting curve did not distort the data fitting, either. To examine the results in Table 6, we found that the sum of squared residuals (SSR) of the EGA is less than that of the others, indicating that the performance of the EGA surpasses its competitors. Considering the same dataset, it was reported that the fitted frequency by the elemental set method [48] was 0.0873, which once again is close to our frequency estimation.

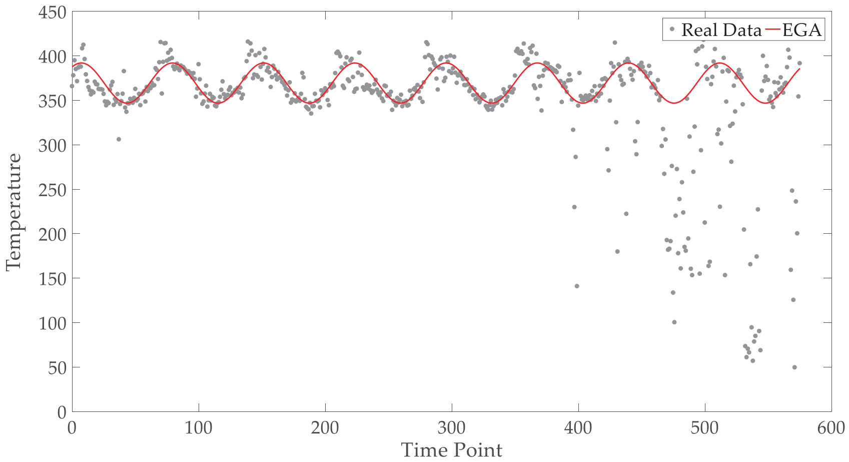


Figure 8. Plots of Circadian Rhythms data and fitting curve.

Table 6. Estimation results of SGA, PSO, CS, CMGA, and EGA for Circadian Rhythms.

Parameter	SGA	PSO	CS	CMGA	EGA
Offset (K)	369.8530	370.1633	370.0567	370.1083	369.4427
Amplitude (A)	23.1916	23.2972	23.3108	23.2656	22.5831
Frequency (ω)	0.0886	0.0891	0.0874	0.875	0.0872
Phase (φ)	0.8699	0.9205	0.9460	0.8877	0.9457
SSR	1.0902×10^3	1.0908×10^3	1.0912×10^3	1.0898×10^3	$\mathbf{1.0194 \times 10^3}$

6. Conclusions

This work presents an EGA for estimating the parameters of sinusoidal signals. The proposed algorithm takes into account the preservation of population diversity, the balance between convergence speed and accuracy, and the implementation complexity. In contrast to other genetic algorithms, the main features of the EGA are:

- a prejudice-free selection mechanism for preserving population diversity;
- a TSC operation for enhancing information exchange among individuals and variables;
- An adaptive mutation strategy to avoid premature convergence and stagnation scenarios.

To evaluate the performance of the proposed method, studies based on benchmark functions are conducted. The results indicate that the EGA not only achieves a higher accuracy and faster convergence speed, but also it provides a good robustness against noise, showing an overwhelming superiority over the SGA, the PSO, the CS, and the CMGA. At last, parameter estimations of real-life sinusoidal signals are performed, which demonstrates the superiority and validity of the proposed algorithm in its entirety.

The EGA has many issues worth studying further, such as the relationship between time consumption and number of offspring, the applications where it excels and fails, and so on.

Author Contributions: Conceptualization, C.J.; formal analysis, M.L.; software, C.J.; Supervision, C.C.; visualization, P.S.; writing—original draft, C.J. and P.S.; writing—review and editing, P.S. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The employed benchmark functions in this work:

$$\text{Sphere: } f_1(x) = \sum_{i=1}^{2} x_i^2 \tag{A1}$$

$$x_i \in [-5.12, 5.12]\,, x_1^* = (0,0), f_1^* = 0$$

$$\text{Zakharov: } f_2(x) = \sum_{i=1}^{2} x_i^2 + \left(\sum_{i=1}^{2} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{2} 0.5 i x_i\right)^4 \tag{A2}$$

$$x_i \in [-5, 10], x_2^* = (0,0), f_2^* = 0$$

$$\text{Rosenbrock: } f_3(x) = \sum_{i=1}^{1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right) \tag{A3}$$

$$x_i \in [-2.048, 2.048], x_3^* = (0,0), f_3^* = 0$$

$$\text{Ackley: } f_4(x) = -20 \cdot e^{\left(-0.2\sqrt{0.5\sum_{i=1}^{2} x_i^2}\right)} - e^{\left(0.5\sum_{i=1}^{2}\cos 2\pi x_i\right)} + 20 + e \tag{A4}$$

$$x_i \in [-32, 32], x_4^* = (0,0), f_4^* = 0$$

$$\text{Griewank: } f_5(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{A5}$$

$$x_i \in [-600, 600], x_5^* = (0,0), f_5^* = 0$$

$$\text{WaveDrop: } f_6(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{2 + 0.5\left(x_1^2 + x_2^2\right)} \tag{A6}$$

$$x_i \in [-5.12, 5.12], x_6^* = (0,0), f_6^* = -1$$

$$\text{X. S. Yang's No.2 fct.: } f_7(x) = \sum_{i=1}^{n} |x_i| \cdot e^{\left(-\sum_{i=1}^{n}\sin\left(x_i^2\right)\right)} \tag{A7}$$

$$x_i \in [-2\pi, 2\pi], x_7^* = (0,0), f_7^* = 0$$

$$\text{X. S. Yang's No.5 fct.: } f_8(x) = \left|1 - e^{\left(-\sum_{i=1}^{n} U_i x_i^2\right)}\right| + \sum_{i=1}^{n} U_i \sin^2\left(2\pi n x_i\right) \tag{A8}$$

$$U_i \sim Unif[0,1], x_i \in [-2\pi, 2\pi], x_8^* = (0,0), f_8^* = 0$$

References

1. Drugman, T.; Stylianou, Y. Maximum Voiced Frequency Estimation: Exploiting Amplitude and Phase Spectra. *IEEE Signal Process. Lett.* **2014**, *21*, 1230–1234. [CrossRef]
2. Beltran-Carbajal, F.; Silva-Navarro, G. A fast parametric estimation approach of signals with multiple frequency harmonics. *Electr. Power Syst. Res.* **2017**, *144*, 157–162. [CrossRef]
3. Liu, B.; Zhang, C.; Ji, X.; Chen, J.; Han, T. An Improved Performance Frequency Estimation Algorithm for Passive Wireless SAW Resonant Sensors. *Sensors* **2014**, *14*, 22261–22273. [CrossRef] [PubMed]
4. Stoica, P.; Li, H.; Li, J. Amplitude estimation of sinusoidal signals: Survey, new results, and an application. *IEEE Trans. Signal Process.* **2000**, *48*, 338–352. [CrossRef]

5. Duda, K.; Barczentewicz, S. Interpolated DFT for $\sin^{\alpha}(x)$ Windows. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 754–760. [CrossRef]
6. Djukanović, S. An Accurate Method for Frequency Estimation of a Real Sinusoid. *IEEE Signal Process. Lett.* **2016**, *23*, 915–918. [CrossRef]
7. Zhang, J.; Wen, H.; Teng, Z.; Martinek, R.; Bilik, P. Power system dynamic frequency measurement based on novel interpolated STFT algorithm. *Adv. Electr. Electron. Eng.* **2017**, *15*. [CrossRef]
8. Ye, S.; Kocherry, D.L.; Aboutanios, E. A novel algorithm for the estimation of the parameters of a real sinusoid in noise. In Proceedings of the 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 2271–2275. [CrossRef]
9. Karimi-Ghartemani, M.; Iravani, M.R. Robust and frequency-adaptive measurement of peak value. *IEEE Trans. Power Deliv.* **2004**, *19*, 481–489. [CrossRef]
10. Samarah, A. A comparative study of single phase grid connected phase looked loop algorithms. *Jordan J. Mech. Ind. Eng.* **2017**, *11*, 185–194.
11. Spavieri, G.; Ferreira, R.T.; Fernandes, R.A.; Lage, G.G.; Barbosa, D.; Oleskovicz, M. Particle Swarm Optimization-based approach for parameterization of power capacitor models fed by harmonic voltages. *Appl. Soft Comput.* **2017**, *56*, 55–64. [CrossRef]
12. Martinez-Ayala, E.; Ayala-Ramirez, V.; Sanchez-Yanez, R.E. Noisy signal parameter identification using Particle Swarm Optimization. In Proceedings of the IEEE 21st International Conference on Electrical Communications and Computers (CONIELECOMP), San Andres Cholula, Mexico, 28 February–2 March 2011; pp. 142–146. [CrossRef]
13. Wang, Y.; Wang, Z.; Zhao, B.; Xu, L. Parameters estimation of sinusoidal frequency modulation signal with application in synthetic aperture radar imaging. *J. Appl. Remote Sens.* **2016**, *10*, 020502. [CrossRef]
14. Kiran, M.S. Particle swarm optimization with a new update mechanism. *Appl. Soft Comput.* **2017**, *60*, 670–678. [CrossRef]
15. Xiao, X.; Lai, J.H.; Wang, C.D. Parameter estimation of the exponentially damped sinusoids signal using a specific neural network. *Neurocomputing* **2014**, *143*, 331–338. [CrossRef]
16. Mitra, A.; Kundu, D.; Agrawal, G. Frequency estimation of undamped exponential signals using genetic algorithms. *Comput. Stat. Data Anal.* **2006**, *51*, 1965–1985. [CrossRef]
17. Djurović, I.; Simeunović, M.; Lutovac, B. Are genetic algorithms useful for the parameter estimation of FM signals? *Digit. Signal Process.* **2012**, *22*, 1137–1144. [CrossRef]
18. Silva, R.P.M.d.; Delbem, A.C.B.; Coury, D.V. Genetic algorithms applied to phasor estimation and frequency tracking in PMU development. *Int. J. Electr. Power Energy Syst.* **2013**, *44*, 921–929. [CrossRef]
19. Mitra, S.; Mitra, A.; Kundu, D. Genetic algorithm and M-estimator based robust sequential estimation of parameters of nonlinear sinusoidal signals. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 2796–2809. [CrossRef]
20. Coury, D.V.; Silva, R.P.M.d.; Delbem, A.C.B.; Casseb, M.V.G. Programmable logic design of a compact Genetic Algorithm for phasor estimation in real-time. *Electr. Power Syst. Res.* **2014**, *107*, 109–118. [CrossRef]
21. Raja, M.A.Z.; Zameer, A.; Khan, A.U.; Wazwaz, A.M. A new numerical approach to solve Thomas–Fermi model of an atom using bio-inspired heuristics integrated with sequential quadratic programming. *SpringerPlus* **2016**, *5*, 1400. [CrossRef]
22. Raja, M.A.Z.; Niazi, S.A.; Butt, S.A. An intelligent computing technique to analyze the vibrational dynamics of rotating electrical machine. *Neurocomputing* **2017**, *219*, 280–299. [CrossRef]
23. Dong, H.; Li, T.; Ding, R.; Sun, J. A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Appl. Soft Comput.* **2018**, *65*, 33–46. [CrossRef]
24. Wang, J.; Ersoy, O.K.; He, M.; Wang, F. Multi-offspring genetic algorithm and its application to the traveling salesman problem. *Appl. Soft Comput.* **2016**, *43*, 415–423. [CrossRef]
25. Sun, Z.; Wang, N.; Bi, Y.; Srinivasan, D. Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm. *Energy* **2015**, *90*, 1334–1341. [CrossRef]
26. Renczes, B. Numerical Problems of Sine fitting Algorithms. Ph.D. Thesis, Budapest University of Technology and Economics, Budapest, Hungary, 2017.
27. Draa, A.; Bouzoubia, S.; Boukhalfa, I. A sinusoidal differential evolution algorithm for numerical optimisation. *Appl. Soft Comput.* **2015**, *27*, 99–126. [CrossRef]

28. Zhang, X.M. Parameter estimation of shallow wave equation via cuckoo search. *Neural Comput. Appl.* **2017**, *28*, 4047–4059. [CrossRef]
29. Merkle, D.; Middendorf, M. Swarm intelligence and signal processing [DSP Exploratory]. *IEEE Signal Process. Mag.* **2008**, *25*, 152–158. [CrossRef]
30. Zang, W.; Ren, L.; Zhang, W.; Liu, X. A cloud model based DNA genetic algorithm for numerical optimization problems. *Future Gener. Comput. Syst.* **2018**, *81*, 465–477. [CrossRef]
31. Raja, M.A.Z.; Umar, M.; Sabir, Z.; Khan, J.A.; Baleanu, D. A new stochastic computing paradigm for the dynamics of nonlinear singular heat conduction model of the human head. *Eur. Phys. J. Plus* **2018**, *133*, 364. [CrossRef]
32. Raja, M.A.Z.; Shah, Z.; Manzar, M.A.; Ahmad, I.; Awais, M.; Baleanu, D. A new stochastic computing paradigm for nonlinear Painlevé II systems in applications of random matrix theory. *Eur. Phys. J. Plus* **2018**, *133*, 254. [CrossRef]
33. Sabir, Z.; Manzar, M.A.; Raja, M.A.Z.; Sheraz, M.; Wazwaz, A.M. Neuro-heuristics for nonlinear singular Thomas-Fermi systems. *Appl. Soft Comput.* **2018**, *65*, 152–169. [CrossRef]
34. Ali, M.Z.; Awad, N.H.; Suganthan, P.N.; Shatnawi, A.M.; Reynolds, R.G. An improved class of real-coded Genetic Algorithms for numerical optimization. *Neurocomputing* **2018**, *275*, 155–166. [CrossRef]
35. Zaman, F.; Qureshi, I.M. 5D parameter estimation of near-field sources using hybrid evolutionary computational techniques. *Sci. World J.* **2014**, *2014*. [CrossRef]
36. Bacaër, N. *A Short History of Mathematical Population Dynamics*; Springer Science & Business Media: Berlin, Germany, 2011. [CrossRef]
37. Sarmady, S. An investigation on genetic algorithm parameters. *Sch. Comput. Sci. Univ. Sains Malays.* **2007**, *126*.
38. Sathya, S.S.; Kuppuswami, S. Analysing the migration effects in nomadic genetic algorithm. *Int. J. Adapt. Innov. Syst.* **2010**, *1*, 158–170. [CrossRef]
39. Sathya, S.S.; Radhika, M. Convergence of nomadic genetic algorithm on benchmark mathematical functions. *Appl. Soft Comput.* **2013**, *13*, 2759–2766. [CrossRef]
40. Umbarkar, A.; Sheth, P. Crossover operators in genetic algorithms: A review. *ICTACT J. Soft Comput.* **2015**, *6*. [CrossRef]
41. Lim, S.M.; Sultan, A.B.M.; Sulaiman, M.N.; Mustapha, A.; Leong, K. Crossover and mutation operators of genetic algorithms. *Int. J. Mach. Learn. Comput.* **2017**, *7*, 9–12. [CrossRef]
42. Wang, K.; Wang, N. A protein inspired RNA genetic algorithm for parameter estimation in hydrocracking of heavy oil. *Chem. Eng. J.* **2011**, *167*, 228–239. [CrossRef]
43. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194. [CrossRef]
44. Nabaei, A.; Hamian, M.; Parsaei, M.R.; Safdari, R.; Samad-Soltani, T.; Zarrabi, H.; Ghassemi, A. Topologies and performance of intelligent algorithms: A comprehensive review. *Artif. Intell. Rev.* **2018**, *49*, 79–103. [CrossRef]
45. Oliver, W.; Yu, J.; Metois, E. The Singing Tree:: Design of an Interactive Musical Interface. In Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97), Amsterdam, The Netherlands, 18–20 August 1997; pp. 261–264. [CrossRef]
46. Andrews, D.; Herzberg, A. *A Collection of Problems from Many Fields for the Student and Research Worker*; Springer: Berlin, Germany, 1985. [CrossRef]
47. Smyth, G.K. Employing Symmetry Constraints for Improved Frequency Estimation by Eigenanalysis Methods. *Technometrics* **2000**, *42*, 277–289. [CrossRef]
48. Smyth, G.K.; Hawkins, D.M. Robust Frequency Estimation Using Elemental Sets. *J. Comput. Graph. Stat.* **2000**, *9*, 196–214. [CrossRef]