electronics

MDPI

Article

Uni-Cycle Genetic Algorithm as an Adaptation Engine for Wireless Channel Equalizers

Mohammed J. Khafaji * and Maciej Krasicki

Faculty of Computing and Telecommunications, Institute of Radiocommunications, Poznan University of Technology, Polanka 3, 61-131 Poznan, Poland; maciej.krasicki@put.poznan.pl

* Correspondence: mohammed.j.khafaji@doctorate.put.poznan.pl

Abstract: A recently developed adaptive channel equalizer driven by a so-called Uni-Cycle Genetic Algorithm (UCGA) is examined in the paper. The authors consider different initialization strategies of the iterative process and compare UCGA against the reference Recursive Least Squares (RLS) algorithm in terms of Bit Error Rate (BER) vs. Signal to Noise Ratio (SNR) performance and convergence rate of an adaptive channel equalizer. The results display a reasonable performance gain of UCGA over RLS for most of wireless channel models studied in the paper. Additionally, UCGA is capable of boosting the equalizer convergence. Thus, it can be considered a promising candidate for the future adaptive wireless channel equalizer.

Keywords: adaptive equalization; genetic algorithm; recursive least squares

check for updates

Citation: Khafaji, M.J.; Krasicki, M. Uni-Cycle Genetic Algorithm as an Adaptation Engine for Wireless Channel Equalizers. *Electronics* **2022**, *11*, 171. https://doi.org/10.3390/electronics11020171

Academic Editor: Cheng-Chi Lee

Received: 2 December 2021
Accepted: 31 December 2021
Published: 6 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Adaptation refers to a chain of steps needed by the filter to adjust its parameters (coefficients) in accordance with the input data so that its characteristics are optimized [1]. Adaptive equalizers create a subgroup of adaptive filters. The area of adaptive equalization has found wide usage in all forms of communication systems. In a communication system transmitting over a wireless channel, the adaptive equalizer is applied to compensate for the harmful effects of multipath transmission on the received signal quality [2]. Adaptation or learning algorithm is one of the characteristics that define the adaptive equalizer, i.e., it specifies how the coefficients are adjusted from one time instant to the next. RLS exemplifies one of the well-known adaptation algorithms used to tune adaptive equalizer coefficients. RLS algorithm possesses a fast convergence rate, which is measured by the number of adaptation cycles (iterations) necessary to accomplish a solution exhibiting satisfactorily low mean square error (MSE). A disadvantage of RLS is high computational complexity [3].

A couple of years ago, evolutionary algorithms, such as the GA, gained researchers' attention as an optimization framework in the field of wireless communication systems [4,5]. Giving consideration to equalization, Humaidi et al. [6] introduced an integrated learning algorithm by hybridizing the standard Least Mean Square (LMS) with the GA; the algorithm is capable of tuning both Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The GA-based equalizer for a wireless communication system that incorporates direct sequence ultra-wideband (DS-UWB) signal spreading has been considered in [7]. In [8], GA was used to estimate the coefficients of high-order IIR filters by applying a time-frequency fitness function. Chen et al. [9] combined a so-called micro-GA (i.e., a version of GA with small population size assumed) with the Viterbi algorithm. As a result, data estimation is conducted jointly with channel estimation. However, the method does not allow channel tracking; it seems clear that the channel equalizer from [9] would require periodical injections of training sequences for re-tuning.

Undoubtedly, the GA has been known for years, but that does not dismiss this method. On the contrary, we believe it is still worth being developed and improved. The GA employs

a population search strategy and utilizes probabilistic transition rules. The search directions are controlled by the objective function and corresponding fitness levels. Therefore, the GA is quite good at navigating through broad search spaces and can quickly reach the globally optimal solution to the optimization problem [10].

The UCGA, considered herein, has been introduced in [11] as an adaptation engine for linear channel equalizers. In the cited work, the GA considers just one generation per one signaling interval, which seems to be enough for efficient channel tracking—a key feature of the systems transmitting over nonstationary radio channels. Accordingly, the optimal settings of the equalizer's taps continue on changing in the sequel of the change of channel conditions.

The UCGA computational cost is significantly smaller than that of the regular GA. (The original GA must process hundreds of generations per one signaling interval before it can output a vector containing the optimal equalizer coefficients).

From [11], it can be concluded that in the performance contest between the popular LMS-based equalizer and the considered UCGA-based equalizer, the latter is an undisputed winner. In the current research, we consider RLS as the reference equalizer algorithm instead of LMS. Additionally, we explore an alternative method of the UCGA initialization and check if the single-1 initialization, introduced hereinafter, can add any gain to the system performance.

The remaining part of the paper is organized as follows: In Section 2, the system model and the tools, used in the current research, are introduced. Section 3 describes the UCGA details. The simulation experiment assumptions are specified, and the results are displayed in Section 4. Section 5 discusses the results. Finally, a brief conclusion is drawn in Section 6.

The following notation convention is used throughout the paper: regular lowercase or uppercase letters represent scalars. Vectors are represented by lowercase bold letters, such as $\mathbf{b}$. If a vector is time-dependant, the respective signaling interval is specified in a bracket, i.e., $\mathbf{b}(n)$. $\mathbb{R}$ represents a set of real numbers. $(\cdot)^{\Re}$ and $(\cdot)^{\Im}$ stand for the real- and imaginary part of a complex scalar, respectively, $|\cdot|$ means the modulus of a complex scalar, and $\lfloor \cdot \rfloor$ returns the largest integer less than or equal to a real argument.

2. Materials and Methods

The performance of the UCGA is evaluated by means of a computer-based simulation experiment. The software used during the study is the GA solver, which is the part of MATLAB packet. The researcher's role is to input the parameters, constraints and, crucially, to properly specify the objective function.

To understand the GA solver configuration in the considered optimization problem, we focus on the system model illustrated by the block diagram from Figure 1. The designed equalizer acquires the samples of Binary Phase-Shift Keying (BPSK) modulated signal, transmitted over a time-varying Rayleigh fading channel. The resulting channel output $\mathbf{r}$ is processed by a transversal filter, i.e., adaptive linear equalizer, responsible for providing estimates of the transmitted symbols, $\hat{\mathbf{r}}$. In detail, the estimate of the nth symbol of a data frame is defined as

$$\hat{r}(n) = \sum_{l=1}^{L} c_l(n) \cdot r(n + l - \lfloor L/2 \rfloor) \quad (1)$$

where L is the equalizer length, and the complex-valued numbers $c_1(n) \cdots c_L(n)$, i.e., the elements of vector $\mathbf{c}(n)$ from Figure 1, represent the tap coefficients of the equalizer in respective signaling interval.

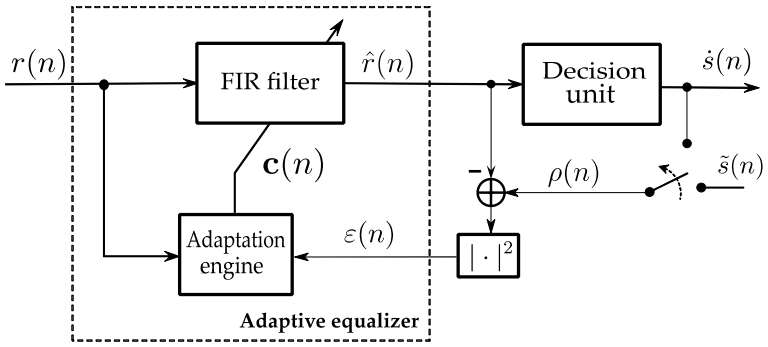


Figure 1. Adaptive Equalizer block diagram.

The adaptation is carried out by observing the error, i.e., the difference between the desired sequence and the equalizer output. The error volume controls the direction of the coefficients adjustment from one iteration to the next in order to approach the optimum set of values. The optimization procedure that the adaptive algorithm utilizes to update the coefficient values is based on the Mean Square Error criterion [12]; thus, the error signal is defined as

$$\varepsilon(n) = |\rho(n) - \hat{r}(n)|^2 \tag{2}$$

During a training phase, a training sequence $\tilde{\mathbf{s}}$ is transmitted. The sequence is known in advance at the receiver side ($\rho(n) = \tilde{s}(n)$), so that the signal received in the training phase can be taken as a reference. The optimization criterion for the adaptive equalizer is to minimize the error (2) according to the MSE criterion. In other words, in the nth signaling interval the equalizer looks for the best vector $\mathbf{c}(n)$ of filter taps.

Having completed the training phase, the adaptive equalizer starts operating on the basis of decision-directed rule. The signal outputted by the filter is regenerated by a decision unit, so that it becomes an ideal BPSK symbol, i.e.,

$$\rho(n) = \dot{s}(n) = \arg\min_{s\in\{-1,+1\}} |s - \hat{r}(n)|^2.$$

Thanks to the reference samples received from the decision unit, the adaptive equalizer can track the statistical variations of the channel without wasting the bandwidth for passing any midamble through the channel. However, it might fail if the decision is made in favour of a BPSK symbol opposite to the actually transmitted symbol.

3. Uni-Cycle Genetic Algorithm

The genetic Algorithm is an iterative technique that simulates the natural evolution process by applying the methods of selection, crossover, and mutation of the "genetic material" according to some probabilistic rules [13]. The "genetic material" (called chromosome) represents the details of the given optimization problem solution, called individual. Several individuals co-exist and constitute a population.

The conventional genetic algorithm has been designed to solve optimization problems that do not evolve over time [14,15]. By the application of various GA operators, the structure of the population continuously evolves (the individuals whose chromosomes exhibit the poorest fitness die, the newborns constitute a new generation; there are also some random mutations applied to the chromosomes to apply some minor random changes). By creating a new generation, the algorithm takes a single step on its path towards the optimal (fixed) solution.

In the case of channel equalizer task, the role of the equalizer is to continuously track the current channel state, so there is no one fixed optimization goal. Conceptually, it is possible to consider several GA generations in order to pursue a solution (the channel taps or, equivalently, the decision on one transmitted symbol) that is optimal for a single signaling interval, and then start from scratch to consider the next signaling interval. Such

a solution would be similar to the one presented in [9], (The reference solution from [9] is even more difficult, as it incorporates the decoder, additionally.).

According to the proposed UCGA strategy, only one generation per one signaling interval is assumed to limit the computational payload. The resultant population from one signaling interval is considered the initial population for the next UCGA instance, related to the next signaling interval, and so on, as shown in Figure 2.

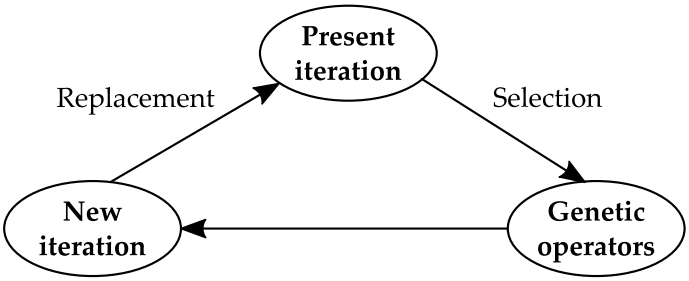


Figure 2. Uni-Cycle Genetic Algorithm framework. The GA operators are the base for the transition from one iteration to the next.

As the channel evolution is rather slow, it can be considered as if the global optimum were fixed over a few subsequent signaling intervals which allows for the UCGA convergence. Obviously, the algorithm cannot find the exact optimum, which is slinking, but a rough estimation of the channel coefficients is generally sufficient to compensate for the channel effect efficiently. The cycle of the UCGA operations gives the chance to adapt the current set of suggested solutions to evolving channel state.

The implementation details of the UCGA are as follows.

- Problem representation

The chromosomes have the meaning of the equalizer coefficients to be optimized. They are represented by vectors containing real numbers. The data type used during the experiment is floating-point [14]. (It is noteworthy that in general one individual may comprise more than one chromosome; in the current system, the individual is determined by exactly one chromosome). In detail, the chromosome represents both real and imaginary parts of the candidate coefficients:

$$\hat{\mathbf{c}} = \left[\hat{c}_1^{\Re} \cdots \hat{c}_L^{\Re}\ \hat{c}_1^{\Im} \cdots \hat{c}_L^{\Im}\right] \tag{3}$$

where $\hat{c}_l^{\Re}, \hat{c}_l^{\Im} \in \mathbb{R}$ define the candidate coefficient $\hat{c}_l$ of the lth equalizer tap (or more precisely, its real and imaginary part, respectively).

- Initialization

The UCGA initialization is executed in the 1st signaling interval. Two strategies are considered. According to the first approach, all the chromosomes are generated randomly; the values of either real or imaginary parts are constrained to the range from -1 to 1. The second strategy assumes random generation of all chromosomes (as above) with the exception for the first individual, for which $\hat{c}_{\lfloor L/2 \rfloor}^{\Re} = 1$, $\hat{c}_{l \neq \lfloor L/2 \rfloor}^{\Re} = 0$ and $\hat{c}_{l=1\ldots L}^{\Im} = 0$. Such settings would be ideal for a single-path transmission, but we expect it to boost the GA convergence even in the case of multipath Rayleigh fading channel.

The equalizer starts to adjust the equalizer taps immediately after receiving the first sample from the channel. It is the typical approach for the initialization of adaptive channel equalizers.

- Fitness evaluation

In every nth iteration (signaling interval), the UCGA population, $\Omega(n)$, consists of P individuals. As mentioned in Section 2, the optimization criterion for the proposed scheme

assumes minimization of the mean-square value of the error signal, given by Equation (2) (the MSE criterion). Thus, to evaluate the fitness of a given individual (represented by chromosome $\hat{\mathbf{c}}$) in iteration n, we use the following cost function:

$$J_{\hat{\mathbf{c}}}(n) = \left| \rho(n) - \sum_{l=1}^{L} \hat{c}_l(n) \cdot r(n + l - \lfloor L/2 \rfloor) \right|^2, \tag{4}$$

where $\hat{c}_l(n) = \hat{c}_l^{\Re} + j\hat{c}_l^{\Im}$. Let us denote the chromosome exhibiting the best fitness in iteration n as

$$\hat{\mathbf{c}}^{\text{best}}(n) = \arg\min_{\hat{\mathbf{c}}} J_{\hat{\mathbf{c}}}(n) \tag{5}$$

In every nth iteration, the UCGA searches for the minimum estimation error

$$J^{\min}(n) = \min_{\hat{\mathbf{a}} \in \Omega(n)} \left(J_{\hat{\mathbf{a}}}(n) \right) \tag{6}$$

across the entire population $\Omega(n)$ to obtain a reference fitness value for comparisons between individual chromosomes; starting from the 2nd iteration, the worst chromosomes are replaced with newborns, being the result of genetic operations like selection, crossover, and mutation.

- Selection

In the current work, selection routine utilizes a method called roulette wheel, i.e., the probability that a given individual becomes a parent is proportional to its fitness [14]. Accordingly, the selection probability in the nth iteration for an individual with chromosome $\hat{\mathbf{c}}$ can be defined as

$$S_{\hat{\mathbf{c}}}(n) = \frac{J_{\hat{\mathbf{c}}}(n)}{\sum_{\hat{\mathbf{a}} \in \Omega(n)} J_{\hat{\mathbf{a}}}(n)} \tag{7}$$

To avoid the loss of the best genetic material, the elite strategy [15] is employed. It means that some high-ranked individuals (in terms of the fitness value) co-exist with the offspring generation. The number of such elite individuals is called an elite count.

- Crossover and Mutation

The intermediate crossover is employed for the UCGA model to merge genes from two parents' chromosomes in the current iteration and produce a new child.

The mutation operation consists of adding a random complex zero-mean Gaussian number A to the chromosome value. The standard deviation of A has been set to an appropriate value, specified in Table 1, after some preliminary experiments.

Table 1. GA settings.

Parameter	Value
Population Size	200
Number of generations	1
Elite count	72
Selection method	Roulette wheel
Crossover type	Intermediate
Crossover probability	0.85
Mutation method	Gaussian
Std. deviation of mutation probability	0.06

Thanks to the mutation, which brings some random changes to the chromosomes, the UCGA is unlikely to stick at one of the local optima. Neither elite individuals nor crossover children undergo mutation procedure.

4. Results

To evaluate the performance of the proposed UCGA-based Adaptive Equalizer (UCGA-AE), the measures of the following characteristics are taken: BER against SNR, and MSE vs. iteration number. Afterwards, they are compared with the reference results obtained by means of RLS-based Adaptive Equalizer (RLS-AE). The UCGA initialization is also investigated. To ensure comparison fairness, the same channel conditions are considered for the proposed and reference methods, including the same random generator seed when simulating the channel performance.

4.1. Simulation Scenarios

Two simple but realistic GSM propagation models with 6 and 12 delay taps, respectively, accurate for an urban area, are considered [16]. The power delay profiles of such channels are presented in Tables 2 and 3, respectively. In order to get a wider scope on the UCGA-AE capabilities, a 6-tap hilly-terrain model, characterized by the power delay profile from Table 4 is also taken into account. The sampling rate of 1.5 MHz is assumed. Respective sample time of ca. 0.67 µs, compared with all power delay profiles, guarantees a selective kind of fading for every considered channel profile. The impact of the Doppler effect on the equalizer performance is studied for every channel model; numerous Doppler shift values are assumed one by one: $f_d = 0,\ 3,\ 15,\ 50,\ 200$ Hz.

Table 2. Typical Urban Area 6-Tap Channel Power Delay Profile.

Tap Number	Relative Time (µs)	Value
1	0.0	−3.0
2	0.2	0.0
3	0.5	−2.0
4	1.6	−6.0
5	2.3	−8.0
6	5.0	−10.0

Table 3. Typical Urban Area 12-Tap Channel Power Delay Profile.

Tap Number	Relative Time (µs)	Value
1	0.0	−4.0
2	0.1	−3.0
3	0.3	0.0
4	0.5	−2.6
5	0.8	−3.0
6	1.1	−5.0
7	1.3	−7.0
8	1.7	−5.0
9	2.3	−6.5
10	3.1	−8.6
11	3.2	−11.0
12	5.0	−10.0

Table 4. Typical Hilly Terrain 6-Tap Channel Power Delay Profile.

Tap Number	Relative Time (µs)	Value
1	0.0	0.0
2	0.1	−1.5
3	0.3	−4.5
4	0.5	−7.5
5	15.0	−8.0
6	17.2	−17.7

The receiver in any real communication system is not capable of predicting exact channel parameters, e.g., time dispersion. Hence, the number of equalizer taps is assumed to be 15 for both the UCGA-AE and the RLS-AE, regardless of the considered channel model. To ensure the accuracy of the results, the data frame contains 100,000 uncoded BPSK symbols from which 500 initial symbols belong to the training sequence.

Table 1 presents the most important settings of the MATLAB GA solver, used to evaluate the performance of UCGA-AE. Regarding the reference RLS algorithm, some preliminary tests have been carried out to find the most accurate value of the forgetting factor—it is set to 0.95.

4.2. Experimental Results

Figure 3 present the performance results for both adaptive equalizers to be compared (UCGA and RLS) for (a) 6-tap urban, (b) 12-tap urban, and (c) 6-tap hilly-terrain channel models, respectively. Each time, a variety of Doppler shift (f_d) values are considered. The measure points at the SNR scale are distributed every 5 dB, which, in the authors' opinion, is enough to display the trend and the curves' locations to each other. The 95% confidence intervals are presented at data points to prove accuracy of the simulation outcomes; they are obtained according to the method presented in [17].

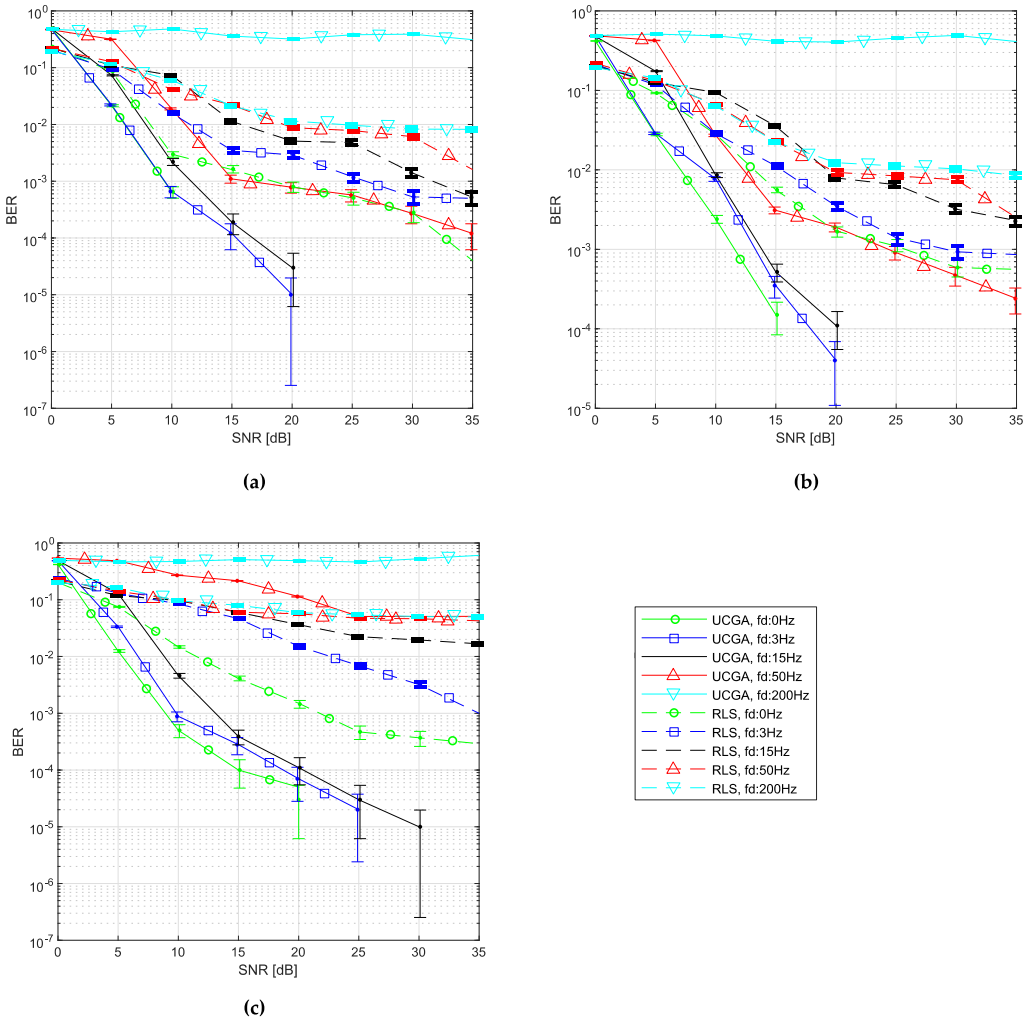


Figure 3. BER vs. SNR performance for the considered adaptive equalizers: GA and RLS under different Doppler shift (f_d = 0, 3, 15, 50, and 200 Hz) over: (**a**) typical urban area 6-tap channel; (**b**) typical urban area 12-tap channel; (**c**) typical hilly-terrain 6-tap channel.

Figure 4 compares different population initialization methods (mentioned in Section 3): fully random initialization, in which all chromosomes in the initial population in the 1st signaling interval are generated randomly, and single-1 initialization. The channel model used in this part of the experiment is the typical urban-area 6-tap model.

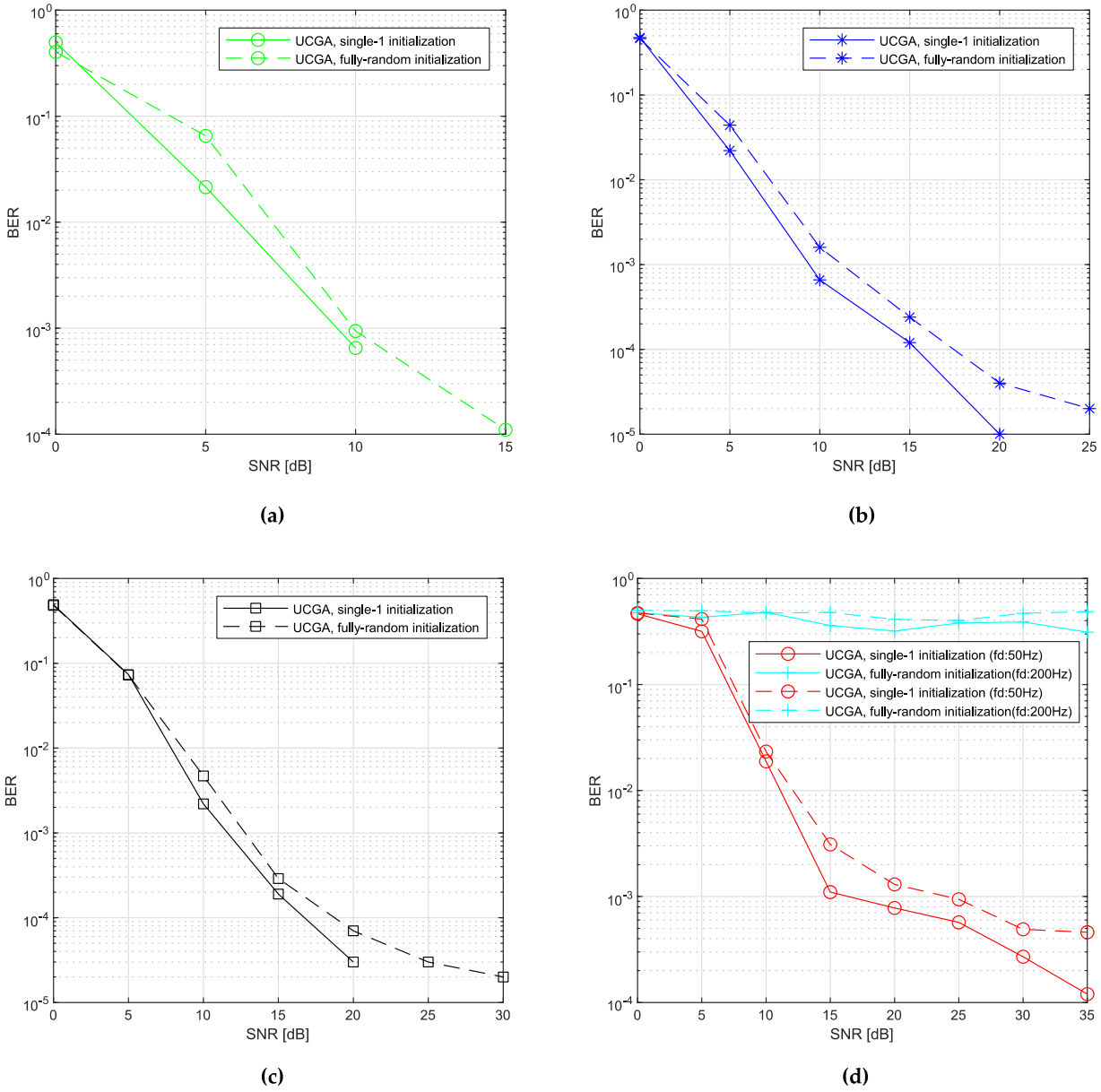


Figure 4. Comparison of different population initialization methods: fully-random initialization and single-1 initialization; assumed typical urban area 6-tap channel model with specific Doppler shifts: (**a**) $f_d = 0$ Hz; (**b**) $f_d = 3$ Hz; (**c**) $f_d = 15$ Hz; (**d**) $f_d = 50$ Hz (red curves) and $f_d = 200$ Hz (blue curves).

The training-phase convergence of both the UCGA-AE and the RLS-AE is also studied for the same channel model, i.e., the curves in Figure 5 represent the MSE vs. iteration number, given SNR = 35 dB and $f_d = 0$ Hz for the considered transmission of a single data frame. More reliable data regarding the training phase convergence are presented in the form of a box plot [18] for every 20th iteration (signaling interval). The box plot, shown in Figure 6, delivers MSE statistics based on the results collected in 500 trial runs (in every trial run only the training sequence is transmitted). The red lines on the boxes

represent the expected values of the median in the considered iteration, the notches delimit 95% confidence intervals of the median. The bottom and the top boxes' ends have the meaning of the 1st and the 3rd MSE quartile, respectively. The whiskers, set to be 1.5 times the interquartile range, cover as much as ca. 99.3% ("almost all") entries. The remaining outliers are shown by + markers.

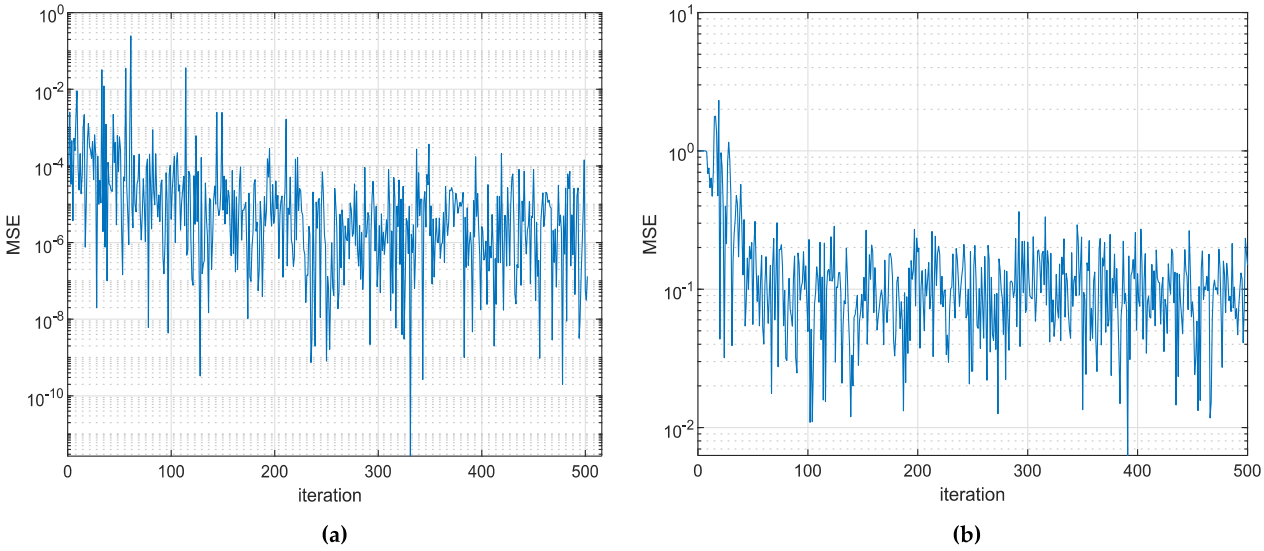


Figure 5. Convergence characteristics at SNR = 35 dB, $f_d = 0$ Hz in the case of typical urban area 6-tap channel: (**a**) UCGA; (**b**) RLS.

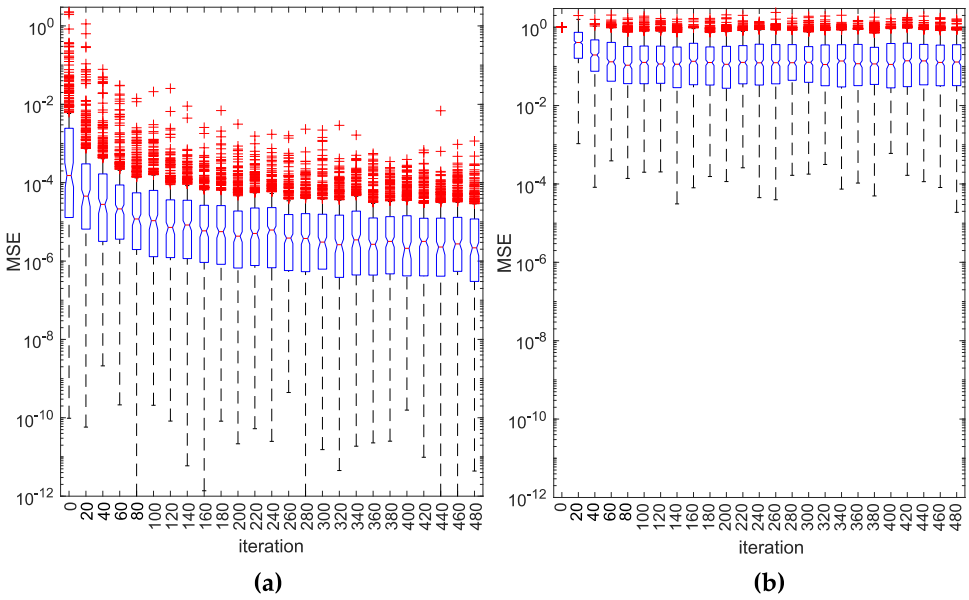


Figure 6. Box plots of convergence characteristics at SNR = 35 dB, $f_d = 0$ Hz in the case of typical urban area 6-tap channel: (**a**) UCGA; (**b**) RLS.

5. Discussion

In general, for all channel models and most Doppler shifts the UCGA-AE (solid lines in Figure 3) is superior to the RLS-AE (dashed lines). A particularly high SNR gain (as much as 17 dB) can be observed at 10^{-3} BER level in the case of either 6-tap or 12-tap urban channel models, if one assumes $f_d = 3$ Hz (refer to Figure 3a,b; note the overlapping curves for UCGA, $f_d = 0$ Hz and UCGA, $f_d = 3$ Hz). Equally remarkable is that in the case of 6-tap hilly-terrain channel (Figure 3c) RLS-AE curves decline much slower with increasing

SNR, in comparison with respective UCGA-AE curves. A weak point of the UCGA-AE is its performance in the case of $f_d = 200$ Hz. Moreover, the efficiency of the proposed UCGA-AE is not satisfactory if the transmission runs over the hilly-terrain 6-tap channel exhibiting 50 Hz Doppler shift.

From Figure 4, it can be seen that the single-1 initialization brings some gain over fully random initialization, i.e., the UCGA-AE is capable of achieving a given BER level at a reasonably lower SNR.

Regarding the convergence characteristics (refer to Figure 5 for single run results and to Figure 6 for full MSE statistics), it must be pointed out that in the considered case UCGA converges faster than RLS as its starting point of the MSE vs. iteration curve is placed significantly lower (ca. 10^{-4} in comparison with 10^{0} for RLS). The average steady-state MSE exhibited by UCGA (ca. 10^{-6}) is unattainable for the RLS (ca. 10^{-1}).

The idea of considering just one generation per signaling interval has a crucial influence on the algorithm computational complexity. This approach makes the cost of the learning part linearly proportional to the population size. Regarding the filtering part, in order to calculate $\hat{r}$ according to Equation (1), given L filter taps, the system requires L multiplications and $L - 1$ additions.

6. Conclusions

The genetic algorithm is a powerful tool for solving numerous optimization problems. The UCGA technique presents the possibility of tracking the channel evolution, which has been verified in a number of simulation experiments, reported in the paper. In the majority of the considered simulation setups, the UCGA-AE remarkably outperforms the reference RLS-AE. As a consequence, the UCGA can be considered a robust channel equalizer engine for the future wireless systems, including 6G telecommunications.

As for any problem solved by means of the GA, it is crucial to properly set the solver parameters and initialize the population. It has been verified experimentally that the single-1 initialization gives faster UCGA convergence than the random assignment of initial values to all chromosomes.

As a future research task, higher-order modulations, such as QPSK, 16-QAM, 64-QAM, etc., could be taken into account. Another item of particular interest is the chromosome representation—fixed point representation would be welcome as the fixed-point arithmetic can be easily implemented on popular logic structures, such as Field-Programmable Gate Arrays (FPGA), or tiny microprocessors, in contrary to floating-point operations.

Author Contributions: Conceptualization, M.K. and M.J.K.; methodology, M.K. and M.J.K.; software, M.J.K.; validation, M.J.K. and M.K.; formal analysis, M.J.K. and M.K.; investigation, M.J.K.; resources, M.J.K.; data curation, M.J.K.; writing—original draft preparation, M.J.K.; writing—review and editing, M.K.; supervision, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Polish Ministry of Education and Science within the bailout for task "New methods of signal transmission and reception, design and testing of radio communication systems" in 2021. The APC was funded by Polish Ministry of Education and Science.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Farhang-Boroujeny, B. *Adaptive Filters: Theory and Applications*, 5th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2013.
2. Rey Vega, L.; Rey, H. *A Rapid Introduction to Adaptive Filtering*; Springer Briefs in Electrical and Computer Engineering; Springer: Berlin/Heidelberg, Germany, 2013.
3. Simon, H. *Adaptive Filter Theory*, 5th ed.; Pearson Education Ltd.: Harlow, UK, 2014.
4. Mehboob, U.; Qadir, J.; Ali, S.; Vasilakos, A. Genetic algorithms in wireless networking: Techniques, applications, and issues. *Soft Comput.* **2016**, *20*, 2467–2501. [CrossRef]
5. Jiang, M.; Hanzo, L. Multiuser MIMO-OFDM for next-generation wireless systems. *Proc. IEEE* **2007**, *95*, 1430–1469. [CrossRef]

6. Humaidi, A.J.; Kasim Ibraheem, I.; Ajel, A.R. A novel adaptive LMS algorithm with genetic search capabilities for system identification of adaptive FIR and IIR filters. *Information* **2019**, *10*, 176. [CrossRef]
7. Surajudeen-Bakinde, N.; Zhu, X.; Gao, J.; Nandi, A.K.; Lin, H. Genetic algorithm based equalizer for ultra-wideband wireless communication systems. *IEICE Trans. Commun.* **2010**, *93*, 2725–2734. [CrossRef]
8. Mostajabi, T.; Poshtan, J. IIR filter design using time and frequency responses by genetic algorithm for system identification. In Proceedings of the 1st International eConference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 13–14 October 2011.
9. Chen, S.; Wu, Y. Maximum likelihood joint channel and data estimation using genetic algorithms. *IEEE Trans. Signal Process* **1998**, *46*, 1469–1473. [CrossRef]
10. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing 53*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 25–48.
11. Khafaji, M.J.; Krasicki, M. Uni-Cycle Genetic Algorithm to Improve the Adaptive Equalizer Performance. *IEEE Commun. Lett.* **2021**, *25*, 3609–3613. [CrossRef]
12. Simon, H. *Digital Communication Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
13. Rothlauf, F. *Representations for Genetic and Evolutionary Algorithms*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 9–32.
14. Sivanandam, S.N.; Deepa, S.N. *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 15–37.
15. Abramson, M.A. *Genetic Algorithm and Direct Search Toolbox User's Guide*; The MathWorks Inc.: Natick, MA, USA, 2007.
16. *3GPP, TS 45.005*; GSM/EDGE Radio Transmission and Reception, Release 14. European Telecommunications Standards Institute: Sophia Antipolis Cedex, France, 2017.
17. Jeruchim, M.C.; Balaban, P.; Sam Shanmugan, K. *Simulation of Communication Systems*; Plenum Press: New York, NY, USA, 1992; pp. 500–501.
18. Documentation on the Use of Boxplots in MATLAB. Available online: https://www.mathworks.com/help/stats/boxplot.html (accessed on 29 December 2021).