Applied Soft Computing 142 (2023) 110323

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Analysis on population-based algorithm optimized filter for non-invasive fECG extraction ☆

Lingping Kong [a,1], Seyedali Mirjalili [b,1], Václav Snášel [a,*,1], Jeng-Shyang Pan [c,1], Akshaya Raj [a,1], Radana Vilimkova Kahankova [a,1], Martinek Radek [a,1]

[a] Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, Ostrava, 70080, Czech Republic
[b] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Australia
[c] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China

ARTICLE INFO

Article history:
Received 31 August 2022
Received in revised form 19 March 2023
Accepted 15 April 2023
Available online 23 April 2023

Dataset link: https://physionet.org

Keywords:
Evolutionary algorithm
Fetal signal extraction
Least mean square filter
Population-based optimization
Independent component analysis
Non-invasive fetal ECG extraction

ABSTRACT

Metaheuristic algorithms (MAs) have become one of the primary tools for optimization in diverse domains, including non-invasive fetal electrocardiogram (fECG) extraction. Research reveals that hyperparameters affect the performance of MAs differently to problems, and some algorithms are problem-specific designed and may produce bad results for different problems. Hence, three questions arise, (1) how much can we trust MAs when solving boxed-constrained fECG extraction problems, and (2) which type of MAs are suitable and adequate for fECG extraction? (3) do MAs find acceptable solutions to a problem that does not formulate the problem comprehensively with an imperfect objective function? This paper focuses on these three inquiries and proposes a framework providing an MA-assisted adaptive filter for non-invasive fECG extraction. The proposed framework has three key components: data pre-processing, MA-based adaptive filter, and post-processing. The pre-processing starts to process the signals detected by pregnant women and extract the desired signal by independent component analysis. Such output signals are then passed to a filter and optimized by population-based algorithms, producing an optimal solution. After that, this solution, as filter weights, will be used for signal extraction and be processed by post-processing. Importantly, the proposed framework is user-friendly that can import any MA algorithm to run and disassemble as a separate assisting tool. This work investigated eight classic and disparate MA algorithms on the Abdominal and Direct Fetal ECG Database (ADFECGDB) dataset, offering comprehensive experimental analysis. We infer from the results that the recordings $r01$, $r02$, $r03$, $r05$, $r08$, and $r09$, where the signals are less noisy and can be solved well. While for recordings $r04$, $r07$, $r06$, and $r10$, the performance can be influenced by hyperparameters for any test algorithms, such as the population size, window size, and other parameters in MAs. Meanwhile, we found that MA may not be the primary target influencing the performance vary.

1. Introduction

Fetal monitoring [1], such as heartbeat signal detection, is crucial for early factors identification that may negatively affect the fetus's health and prevent potential damage. The electrocardiogram (ECG) [2] is an easy, effective, non-invasive investigative method [3] for detecting various heart problems [4]. The Fetal ECG signal reflects the electrical function of the fetal heart and accurately delivers valuable data (fetal heart rate, fHR) on its physiological state.

It is not an easy task for the fECG extraction [5] from the recordings of electrodes placed on the skin of a pregnant mother. On the one hand, the fECG signal is weak compared to the mother's voltage signal, and the overlapping waves of heartbeat in time between mother and fetus arise [6]. On the other hand, the inevitable noise from electromyographic activity affects the obtained recordings. Moreover, the fECG obtained by the electrodes placed on the mother's skin contains the maternal ECG (mECG). Hence canceling the mECG from recordings is a challenge and essential work.

Researchers commit to the study of fECG extraction with techniques [2,7] such as neural networks (NN) [8], transformation in both spaces (time) and frequency [9] (wavelet or Fourier transform), data analysis [10] (principal components analysis,

☆ This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic in project "Metaheuristics Framework for Multi-objective Combinatorial Optimization Problems (META MO-COP)", DST/ INT/ Czech/ P-12/ 2019.

* Corresponding author.
E-mail address: vaclav.snasel@vsb.cz (V. Snášel).

[1] All authors contribute the same.

https://doi.org/10.1016/j.asoc.2023.110323
1568-4946/

Table 1
MA algorithms applied in fetal signal extraction problems, Optional: replaceable of MA without effort; Parameters: *Tuned* for specific algorithm and data, *Default* setting as its (chosen MA) implementation.

Year| Paper	MAs	Optional	Description	Parameters	dataset
2010| [20]	GA	No	GA works on the results after filter	Missing	SISTA/DAISY http://www.esat.kuleuven.ac.be/sista/daisy/
2007| [21]	GA	No	noise cancellation based on real FECG	Tuned	http://www.tsi.enst.fr/icacentral(unreachable)
2012| [22]	GA	No	A tool for training the ANFIS structure	Tuned	PhysioBank Database
2010| [23]	GA	No	GA used to tune parameters in LMS	Tuned	http://www.esat.kuleuven.be/sista/members/biomed/data006.htm(unreachable)
2012| [24]	GA	No	Tune parameters for fuzzy wavelet NN	Missing	subset of MIT-BIT [25]
2015[26]	PSO	No	Tune parameters for extended Kalman smoother	Missing	DaISY database
2018| [27]	DE	No	Tune parameter in LMS	Tuned	subset of Daisy [27]
2021| [28]	MFO	No	Tune parameter in LMS	Tuned	Physionet dataset
2020| [29]	FA[a]	No	Tune the parameters of ELM	Tuned	subset of Daisy dataset
2022| [30]	SA-GWO	No	Tune parameters in adaptive filter	Tuned	*figshare* repository [31]
2022| [32]	MF/FFO	Patial	Tune parameters in LMS	Missing	Missing
Our(Proposed)	MAs	Yes	Auxilary tool for tuning parameters	Default	ADFECGDB

[a]firefly algorithm (FA), extreme learning model (ELM), Moth Flame (MF) And Fruit Fly Optimization (FFO), Grey wolf algorithm (GWO) with Sequential Analysis (SA).

independent component analysis), data separation [11] (blind source separation, source decomposition, adaptive noise cancellation). The NN-based extraction method has superior quality, while it needs extensive data to train the model. Transform-based extraction method significantly scales down the computational cost. The data analysis-based extraction method helps monitor and separate the sources, while the extraction accuracy depends on the biosignal database. Adopting an adaptive filter [12] in the abdominal ECG (aECG) signal provides an efficient and sufficient FECG signal extraction. Besides, diverse methodologies can be available to improve the extraction results' effectiveness on fECG. An adaptive filter [13] adapts its transfer function based on an optimization approach motivated by an error signal. In this regard, researchers have employed various categories of adaptive filters to separate fetal and maternal waves. The corresponding techniques, such as Kalman filter [14], Least mean squares filter, Kernel adaptive filter, Recursive least squares filter [15], and Wiener filter, invariably operate one or more reference maternal signals for either training an adaptive or coordinated filter for the extraction of the fetal heartbeat movement.

The evolutionary algorithm (EA) [16] belongs to a subset of bio-inspired computing. This algorithm solves problems through mechanisms inspired by nature's biological evolution, such as reproduction, mutation, recombination, and selection. In a population, individuals are the potential solutions to the problem, and the problem function value defines the grade of the solutions. The population's evolution occurs after the repeated application of the population updating. Researchers have proposed a lot of popular and efficient EAs, such as Genetic algorithms (GA) [17], Differential evolution (DE) [18], and Particle swarm optimization (PSO) [19].

In early research, Suliman and Zeng [20] proposed to record and denoise abdominal ECG (aECG), then did the subsequent separation of maternal ECG by the adaptive filter. At last, the GA is used for the undesired signals cancellation. Kianoush and Ebadi [21] proposed a fetal signal extraction mathematical model based on a simple template fECG signal, which model was further used as a reference and was important for signal processing in the author's belief. It operated GA to fit the obtained signal from the adaptive filter to the actual fECG template. At the same time, the derived fECG template was constructed by the components of three individual *sinc* functions with specific bias. The objective was to simulate the parameters in the model to eliminate the error between the model signal and the real signal. The authors in paper [22] extended the work [20] by using adaptive neuro-fuzzy inference system (ANFIS) [33] structure to extract the transformation between aECG and mECG. GA was employed as a tool for training the ANFIS structure. The author in paper [23] suggested using a time-varying finite impulse filter (FIR) with GA for adaptive filtering. Instead of initializing the coefficient as weight in the least mean squares filter (LMS) [34], the population initialized in [23] contained a set of filter frequencies ranging from minimum 0.01 Hz and maximum 100 Hz frequency (FIR type bandpass). Furthermore, the objective was to minimize the average quadratic error. The combination of fuzzy wavelet and neural networks [2] was proposed in [24] with the GA for adapting the trainable parameters, such as weights, dilation, and parameters in membership functions.

Recently, The author [27] proposed to apply the DE algorithm for tuning the adaptive filter coefficient, where the reference signals are from signals obtained from the thoracic and abdominal region of the pregnant woman. Similarly, the objective was still to minimize the summation of square errors. Majid [29] presented the method using an extreme learning model (ELM) [35] to train extracted signal by LMS from mECG and aECG and apply the firefly algorithm (FA) [36] for the parameters tuning in ELM. Sequential Analysis (SA) [37] was also used to tune parameters while creating a signal template that matches the mECG. Finally, the author in [32] compared the performance of both flame and fruit fly optimization algorithms on signal extraction of tuning parameters in an adaptive filter.

We summarize the key points of some existing papers in Table 1 with the information of (1) the type of MA, (2) the replaceability of the MA, (3) what is the purpose of the combination with MA, (4) the parameter tuning problem, (5) dataset for the experiment. Table 1 shows several problems that cannot be ignored. (1) Those documented papers did not explain why they chose these MAs, such as GA, DE, GWO, or MF. Are these MAs more suitable than others? (2) The authors proposed to tune parameters for this application problem without illustrating the impact on the performance of their methods. Do the parameters with tiny changes cause bad performance in their method? are the parameters applicable to diverse pregnant women data? As we understand, the clinical medical tool should not be sensitive to the experimental setting, such as the population size of MA, the filter's window size, etc. It is undesirable to tune the parameters of MA for each person or situation, as we do not have a real fetal signal as the comparison to verify the correctness of tuned parameters in a real-life situation. Though the EAs have been studied and stated effective in many fetal signal extraction research works [38], the experimental and analysis are incomplete in the application sense. The algorithms applied above are

population-based and stochastic-based searching methods, which could not reach concrete results without numerous random and parameter-sensitive tests. Furthermore, the parameters concerning optimization algorithms and adaptive filters determine the performance of the signal extraction, which cannot be person-specific tuned due to the individual difference and complexity and is also not realistic.

Hence, we fill the gap of analysis on the performance of a wide range of MA algorithms in this fetal signal extraction area which can inform comprehensive performance details and algorithm development in the future. The main contributions of this work are as follows:

- We develop a fetal signal extraction framework that comprises three components, signal pre-processing, adaptive filter, and post-processing, based on the advantage of blind source separation, adaptive filter, and optimization algorithms techniques.
- We generalize the framework with a plug-in-play interface that can employ *Any* optimization algorithm without effort in assisting the adaptive filter operation, as we implement the fetal extraction filter process as an abstract objective problem.
- This framework aims at eliminating the cumbersome steps of parameter tuning in use. Instead, this framework adopts an algorithm with its default parameter defined in its implementation, making this framework applicable for general usage.
- We design a post-process step to improve the accuracy performance to the level of approval, as the results by MA may be bad due to un-tuned parameters related to the algorithm itself (i.e., crossover rate) or the experiment (i.e., population size).
- Our main goal is to provide the performance analysis that informs the possibility of applying MA and the direction of improvement. Hence we conduct extensive experiments with eight disparate classical algorithms on 10 data and offer detailed results analysis. Furthermore, the comprehensive experimental analysis paves the way for practical population-based signal extraction.

The remainder of this paper is organized as follows. Section 2 reviews the preliminary information and emphasizes the optimization algorithms. Section 3 presents the methodology and the implementation details. The extensive experiments and results are in Section 4. At last, we conclude the paper in Section 5.

2. Preliminaries and essential definitions

As a preliminary, we introduce the most related techniques we used or referred to in this paper, namely independent component analysis (ICA) [39], LMS, and optimization algorithms, highlighting the population optimization algorithms.

2.1. Independent component analysis, ICA

ICA belongs to a Machine Learning (ML) technique section known for solving cocktail party problems (or named Blind source separation). The cocktail party problem depicts a scenario where many people simultaneously talk. In this room, 'n' microphones are placed to record the voice signal from 'n' speakers. The goal of ICA is to separate 'n' independent person's speech (components) from mixed recordings of microphones. Instead of maximizing the variance or mutual orthogonality of voice signals by principal component analysis, ICA focuses on maximizing the mutual independence of the speaker's voice.

ICA assumes the non-Gaussian signal distribution and linear mixtures inside respective source signals. Then the signal separation problem becomes the equation as follows:

$$\mathbf{x} = \mathbf{As} + \sigma, \quad \text{or} \quad \mathbf{x} = \sum_{k=1}^{n} m_k \mathbf{a}_k + \sigma \tag{1}$$

where the observed recording vector $\mathbf{x} = \{x_1, \ldots, x_n\}^T$ associating with basis weights $\mathbf{a}_k = (a_{1,k}, \ldots, a_{n,k})^T$. The mixing weights matrix denotes $\mathbf{A} = \mathbf{a}_1, \ldots, \mathbf{a}_n$, and the independent components as $\mathbf{s} = \{s_1, \ldots, s_n\}^T$ $\mathbf{x}$. σ is considered uncorrelated Gaussian noise.

2.2. Least mean squares, LMS

LMS algorithm is a type of filter in ML developed by Widrow and Hoff (1960) for electrical engineering applications [34]. It follows the Steepest descent principle (while LMS belongs to the stochastic gradient descent method due to the noisy gradient from obtained data) while the estimation is continuous. Hence the LMS algorithm can adapt to changes during the statistics, thus being considered as an adaptive filter [40].

LMS uses autocorrelation matrix $R = X^T X$ and the correlation vector $P^T = d^T X$ to create an algorithm that minimizes $\{|\epsilon|^2\}$. During measurements, it transforms the signal recording from a complete-time series into multi-sampled data X.

$$\epsilon = d - y = d - XW$$

where ϵ denotes the output error, d is the reference data (desired signal), and y notes the simulated production.

The filter weight W updating uses estimated values, and formula[2] as

$$\hat{W}_{n+1} = \hat{W}_n + \mu u_n \epsilon_n, \text{ where } \epsilon_n = d_n - u_n^H \hat{W}_n \tag{2}$$

For the signal extraction example in our case, based on the definition of the gradient, we can write the weight update [41] as

$$\begin{cases} \sum_{k=1}^{n} \epsilon_k^2 & = \sum_{k=1}^{n} d_k^2 - W^T R W - 2P^T W \\ \nabla_k & = -2\epsilon_k X_k^T \\ W_{k+1} & = W_k - \mu \nabla_k = W_k + 2\mu \epsilon_k X_k^T \end{cases} \tag{3}$$

where n denotes the length of signal, k is the index of time sequence, d_k represents the amplitude value of reference signal at time k; and R, X and P are the same as introduced above, μ is the step learning rate. ϵ_k is the output error at time k.

2.3. Optimization algorithm

Optimization is to find the best solution for a given optimization problem among all possible solutions with the ultimate goal of maximizing or minimizing one or multiple objective functions. This function evaluates the quality of the solution, termed fitness value in EAs.

Population-based optimization algorithms (POA) [42] belong to the stochastic optimization class, which maintains a set of individuals (population) and utilizes randomness in the search procedure. Meanwhile, the optimization process deals with solution updating, exploration, and exploitation. By default, each individual is a candidate solution to the problem. Therefore, the population with a pool of candidates increases the likelihood of finding the global optima and bypassing the local optima.

POA is famous for solving challenging objective issues, such as multi-modal problems (multiple optima), constrained problems,

[2] https://www.eit.lth.se/fileadmin/eit/courses/ett042/LEC/notes2.pdf

Table 2
A list of classic and effective single-objective optimization algorithms, Abbr-> Abbreviation.

Abbr.	Full Name	Techniques & Description
OFA [43]	Optimal Foraging Algorithm	uneven opportunities for the individual to capture prey through recruitment strategy &simulate animal Foraging based on Optimal Foraging Theory
FRCG [44]	Function minimization by conjugate gradients	It is a light algorithm using quadratically convergent gradient method to solve the unconstrained minimization problem & Over six thousand citations
DE [45]	Differential Evolution	similar computational steps updating related to parameters, scale factor (F), crossover rate (CR) & traditional and powerful stochastic real-parameter OA
PSO [46]	Particle swarm optimization	particle movement are based on the history of its current and best (best-fitness) locations, with some random perturbations & one of the most attractive research interests in literature.
GA [47]	Genetic algorithm	natural selection with operators such as mutation, crossover and selection & one of the most studied research interests in literature
SHADE [48]	Success-History based Adaptive DE	mutation strategy (current-to-p best/1), an external archive, and adaptive control parameters F, CR (history-based parameter adaptation scheme) & Rank 2,3 places algorithms from CEC2018 are based on SHADE
CMAES [49]	Covariance matrix adaptation-Evolution strategy	self-adaptation with derandomization and cumulation in the mutation distribution & derivative-free methods for numerical optimization of non-linear or non-convex continuous optimization problems[a]
IMODE [50]	Improved multi-operator DE	It emphasizes the best-performing operator from multiple differential evolution operators. & Winner of CEC2020 competition

[a] https://en.wikipedia.org/wiki/CMA-ES.

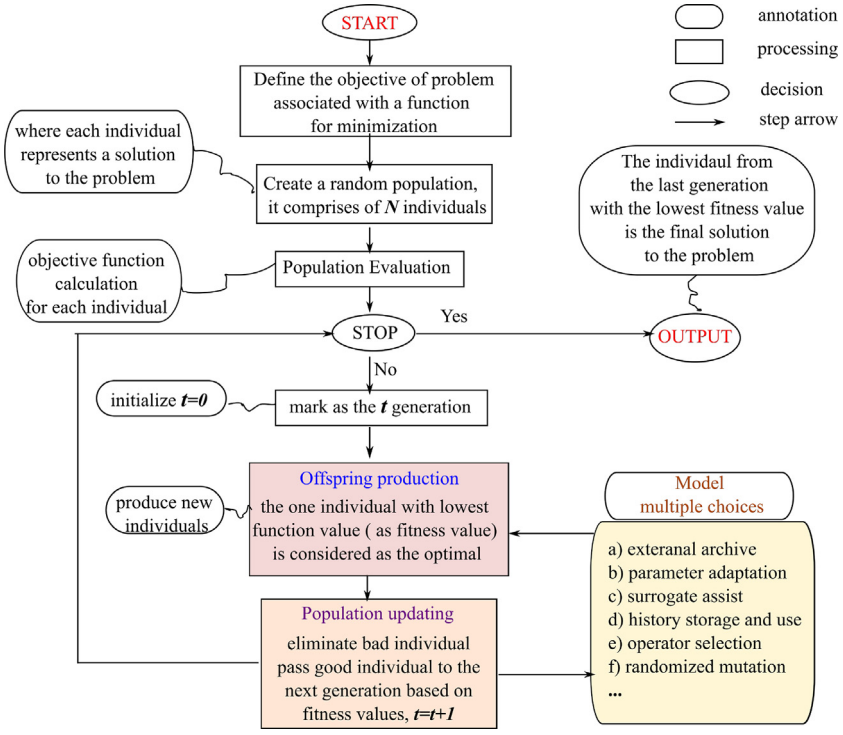


Fig. 1. A simple flowchart of single-objective optimization algorithm.

etc. Examples of POA include GA, DE, and PSO. Table 2 lists some other POAs with algorithms' abbr/full name, techniques procedure, and descriptions. The algorithms are selected roughly by considering computation cost, various nature-inspired theories, gradient-based updates, and complex problem-specific design.

A flowchart of a single-objective optimization algorithm is as Fig. 1.

As shown in Fig. 1, population-based optimization algorithms include four steps initialization, evaluation, offspring production, and updating. First, the population evaluation process calculates the objective function's fitness value (output) according to the individual (input). Then, the population updates positions based on rules at each iteration (a generation). Finally, the optimization process ends when the generation reaches the maximum number or the desired solution is obtained. The population-based optimization algorithms vary in two sub-modules, offspring production, and population updating, where different algorithms design or utilize different techniques.

The LMS algorithm produces the filter coefficients that produce the least mean square of the error signal. In the fECG

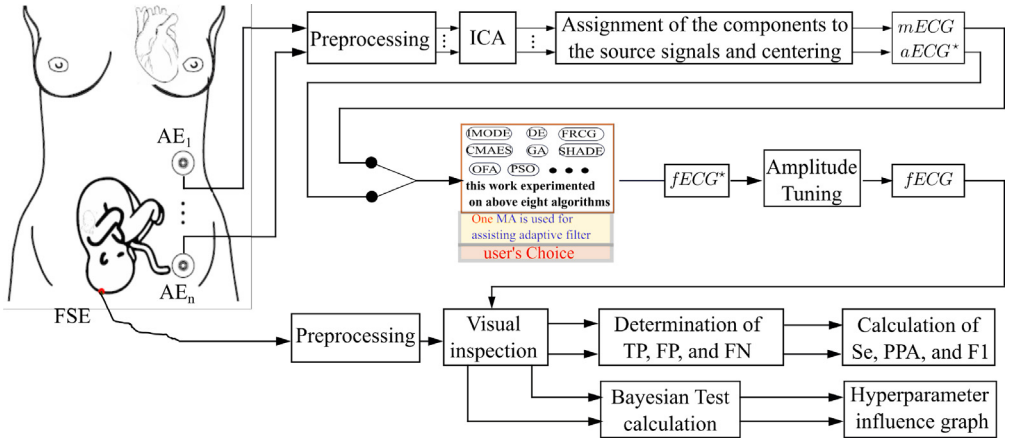


Fig. 2. Pictorial illustration of the proposed framework for fetal signal extraction. The detailed pre-processing is shown in Fig. 3. (Left-side body image is from [51]).

extraction problem, this reference signal is the aECG, and the goal of LMS is to imitate the mECG signal by the coefficients to minimize the error signal (fECG signal). In this work, instead of using LMS to update the coefficients, we use EAs with the evolution process to generate the appropriate coefficient that fulfills the goal of extracting the fECG signal. The details will be introduced in the next Section.

3. Methodology

In this section, we introduce the methodology of proposed fetal signal extraction. First, we briefly introduce the flowchart of the proposed system. Then, we break down the blocks of the system in detail and emphasize the following three parts: data pre-processing, optimization problem selection, and the objective function design.

Fig. 2 describes the proposed fetal signal extraction model, which contains signal extraction modules that sequentially perform on source data and a visual inspection sector; The signal extraction block starts from pre-processing to the signal optimization process. Next, electrodes sense the fetus's ECG waveform transvaginally using an invasive fetal scalp electrode (FSE) method that can only be used during delivery.

3.1. Data pre-processing

The pre-processing module feeds multiple signals detected by electrodes into the Finite impulse response filter (FIR) and chooses certain bands with a bandpass filter for the noise and variation of isolines elimination. Then pass the processed signal into the ICA module for the source separation.

3.1.1. FIR filter processing

The FIR filter was used in the pre-processing phase as it is a well-known tool for bandpass and bandstop filtering and, thus, among the most popular approaches used in fetal ECG processing [52]. The disadvantage of the second widely used method for pre-processing, the IIR filters, is the nonlinear phase response, while the FIR filters are marginally stable and linear [53]. For the same reasons, they are also a common part of adaptive filters.

The setting for FIR includes:

- The lower/upper limit frequency $f_{FIR,L}/f_{FIR,U}$ with 3 Hz and 150 Hz, respectively;
- Filter order 500 N_{FIR};
- The sampling frequency f_s which the one sampling the aECG signal;

Note we assign five recordings of the ADFECGDB database with one kHz sampling frequency and the other five with 500 Hz.

The input signals are derived from four electrodes marked as AE_1 to AE_4, and AE_0 serves as the reference signal. N is the active ground. Fig. 3 presents the electrode layout, and the source information is from the ADFECGDB database.

3.1.2. Source separation

The ICA algorithm was selected as it has proven effective when combined with the adaptive algorithms [54]. As highlighted in [38], the advantage of using the ICA is that it also provides the aECG^{star} signal, which is more suitable as an input than the pure aECG signal. The reason is that in such a signal, the maternal component is already suppressed and the fetal one enhanced; thus, the outcomes of the adaptive filters are also better.

The signal processed by the above step is passed to the ICA module. The setting of ICA includes:

- The number of components, $ICA_n = 3$;
- The convergence criterion $\delta = 1e-5$;
- The maximum number of iteration $k_{ICA} = 100$;

We consider the three components obtained by ICA: mECG, aECG^{star}, and noise. The aECG^{star} signal comprises the enhanced fECG signal, and the amplitude level is on the same scale as the mECG signal. There is no specific order of three components derived from the source, and each cycle causes amplitude alteration due to standardization. Moreover, several samples show time-shifting and rotation on the source components. For this reason, an auto-centering module is used to correct the direction of the component signal. After that, to align the amplitude and time of the mECG component signal based on the mQRS complexes, aECG*. Finally, perform a standardization again during centering automatically or manually when necessary, where a user subjectively picks the mECG signal and the aECG*. The output signals mECG and aECG* have a dimensionless unit. Then the signals pass to the optimization algorithm module for extracting the fECG.

3.2. Optimization algorithm selection

Numerous EAs have been proposed and applied in many different industrial applications. Consequently, the performance of various EAs may vary much concerning the different problems. Therefore, for better analysis and experimentation, we choose the algorithms with diverse evolution updating, which help identify the suitable algorithms for signal extraction problems.

Correspondingly, we choose eight classic algorithms. Namely, OFA [43] FRCG [44], DE [45], PSO [46], GA [47], IMODE [50],

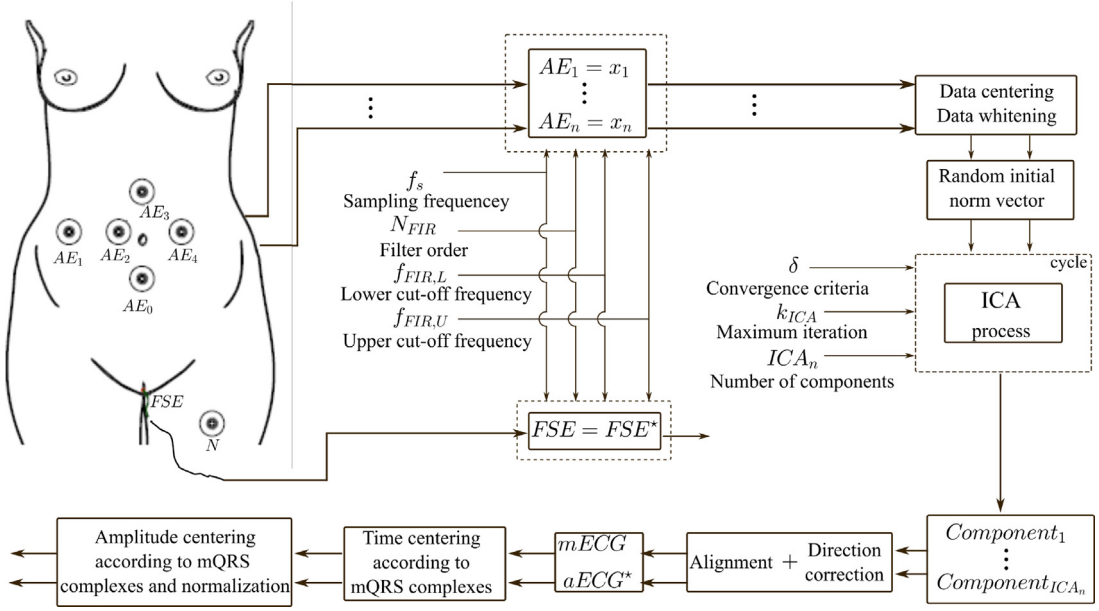


Fig. 3. Signal pre-processing process (first component of the proposed framework): i.e., bandpass filter, ICA algorithm [39] for source separation, and signal rotation, centering, and alignment correction.

SHADE [48], and CMAES [49], for the fetal extraction experiment. OFA belongs to the group of Optimal Foraging. Foraging aims to find resources such as food and shelter without knowing it in advance, which requires the algorithm to proceed with a robust search in sparse and noisy situations. FRCG does not belong to the bio-inspired algorithm based on conjugate gradients from linear algebra. In line-search, the gradient decides the updating directions with two *beta* and *sigma* parameters. GA, PSO, and DE are arguably the most studied and influential classic optimization algorithms. Moreover, these algorithms and their variants contributed to and continue to benefit in the areas related to optimization problems. CMAES has a pretty different evolution strategy from GA. Instead, CMAES uses the known sample solutions and inversely infers the parameter values most likely to lead to such a result. It is similar to the maximum likelihood of setting a model without knowing the parameters. As a result, the performance of CMAES is competitive with other baseline algorithms.

SHADE and IMODE are both inherited from DE. SHADE starts from improving the JADE [55] algorithm, which dramatically improves the DE and adopts a control parameter adaptation mechanism, *current-to-pbest/1* mutation strategy, and an external archive. Based on JADE, SHADE further provides a history-based parameter adaptation scheme, which outperforms many previous DE variants. Furthermore, the experiment shows that various operators do not perform consistently on all problems. IMODE extends the search by cooperating with multiple operators and giving the best-performing operator a more prominent probability.

3.3. Objective function design

From the preliminary background, we learn that the LMS method updates the coefficient (weights ω) based on gradient information of errors. However, due to the noise influence from obtained data, the algorithm adaption to the optima solution is not straight and 100% accurate guaranteed in the absolute sense. This noise issue leads to multiple optional coefficients (results) under different initial weights or step sizes.

Population-based optimization algorithms work similarly for finding the best candidate solutions (the weights in the filter of fetal extraction) based on stochastic search instead of gradient information. However, the well-developed population-based algorithms are more likely to find the global optima and avoid trapping in the local minima. In this work, we use the most straightforward objective function, the summation of a square error on a complete signal.

The optimization objective is to minimize $\{|\epsilon|^2\}$ (indicating the remaining signals of aECG without mECG, aka. considered fECG). During measurements, we first transform the signal recording from a complete-time series into multi-sampled data X.

$$\epsilon = d - y = d - XW = \begin{bmatrix} d_0 \\ \vdots \\ d_{t-1} \end{bmatrix} - \begin{bmatrix} x_{0,1} & \dots & x_{0,m} \\ \vdots & \vdots & \vdots \\ x_{t-1,1} & \dots & x_{t-1,m} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

where ϵ denotes the output error, d is the reference data (desired signal, extracted aECG* by ICA), and y notes the simulated production (estimated desired reference signal), and signal X is constructed from extracted mECG by ICA. t denotes the iterative index of chosen MA. In our case, as in Fig. 4, the filter weight W is fixed and produced by MA (one optional algorithm). For the signal extraction example in our case, based on the definition of the gradient, we can write the weight production as

$$W_{n+1} = p_{best}^{t+1}, \text{ where } p_{best}^{t+1} \in \mathbf{P}^t \text{ (Population/solution of MA)} \quad (4)$$

where P^t denotes the evolution population at t iteration index of MA, and p_{best}^{t+1} is the best solution of P^t.

Note *(1)* The actual objective function is unknown (a black box) for fetal signal extraction. We choose the simplest objective function, the summation of a square error on a complete signal. Though this objective function is imperfect, the experiments show that the results are still acceptable in a proper environmental setting.

Note *(2)* The experiment set a small population size and a small iteration number to speed up the signal extraction process, as, in actual situations, it is desired to acquire the result within a proper time.

Fig. 4 shows the data flow with the input signals. This process comprises the signal window sliding and weights optimization process diagram. The symbols s_t and y_t denote the obtained mECG and aECG signals, respectively. D denotes the window size.

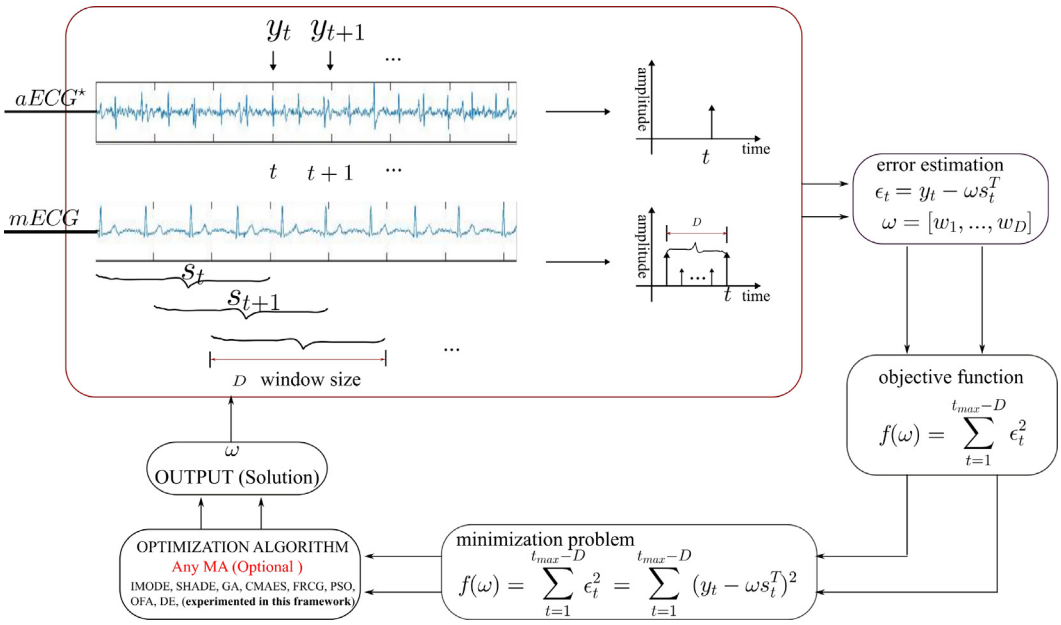


Fig. 4. Adaptive filter assisted by MA; Block diagram of optimization process with the objective function minimize $f(\omega)$.

The ϵ denotes the estimation error. We can consider this error as the extracted fECG. And the coefficient ω in *error estimation* determines the estimation results ϵ, further controlling the *objective function* value $f(\omega)$. At the same time, the ω is the optimization object of an optimization algorithm, where a solution represents a ω, and the best ω is the one with the lowest function value of $f(\omega)$.

The final output of an optimization algorithm is the coefficient ω, and this ω determines a final ϵ sequence $\epsilon_t, \epsilon_{t+1}, \ldots$, then this ϵ sequence is considered as the extracted fetal signal. We use this extracted signal for further processing, e.g., amplitude tuning, then accuracy verification.

Note *(3)*. We adopt a mirror operation on the signal (which flips a short-length signal and adds it to the original signal) to ensure the extracted fetal signal is long enough.

3.4. Post-processing

This framework aims to eliminate the influence of individual-related parameters in each population-based algorithm, as we stated above, trying to generalize the framework and making it applicable to real-life situations. However, each MA individual has its unique evolution characteristics and the default parameter settings from its implementation corresponding to *The* algorithm may not be suitable for the fetal signal extraction problem, leading to a bad result. In addition, common parameters for all MAs, such as population size, window size, or maximum evaluation number, may also affect the performance (and it is indeed, as in Sections 4.3.3 and 4.3.5).

We propose to proceed with an amplitude tuning (AT) process, which adjusts the signal's amplitude, making the potential peak index value (amplitude) of the signal outstanding for detection, where the input data of this AT process is the output of the adaptive filter assisted by MA, e.g., fECG* as in Fig. 2. The AT process output is the final extracted fECG signal, which will then be inspected for signal detection accuracy.

4. Experiment and results

This section details the environmental and experimental setting, including the test platform, experiment preparation, and experimental result & analysis.

4.1. Environmental setting

We manage all selected algorithms embedded in PlatEMO to ease the differences in algorithm implementation, where PlatEMO[3] is a comprehensive MatLab-based platform with over 150+ classic and up-to-date algorithms. We do not change the default individual-related parameters in each algorithm and call each algorithm to solve the abstract problem (implemented in the framework) for ten independent runs. The default individual-related parameters are as what was initially set in PlatEMO. However, there are hyperparameters, such as population size and window size, that a user sets. Those non-individual parameters used in this work are as in Table 3. The MA-assisted filter process is to find the proper weight (solution), and then the obtained weights are fitted as Eq. (3) to produce the desired signal, labeled as fECGG*. At last, the fECGG* is passed to the AT process to get the fECG signal. Then we use the QRS detector [56] to measure the extraction performance of the obtained fECG results.

4.2. Performance metrics and dataset

The evaluation criteria differ from publication to publication. However, there are some metrics commonly found in most the publications, F1, sensitivity (Se), and positive predictivity (PPA)*.[4] Those are related to the calculations of positive values (TP), false-positive values (FP), and false-negative values (FN). TPs and FPs denote the number of detected R-peak on the obtained fECG signal by algorithms, are indeed (exact), falsely (non-exist), or missed R-peak locations determined by actual reference signal, respectively. We would consider an R-peak location as correct detection if it differed from the reference annotation by less than 50 ms.

In most clinical studies, an accuracy of 95% and higher is deemed high in performance, 80% to 95% to be medium in performance, and any parameter lower than 80% is considered low in performance [58]. However, this paper is not a clinical study and focuses on providing a framework for researchers to test

[3] https://github.com/BIMK/PlatEMO

[4] The positive predictivity has different names in literature, as in [5] abbr. as PPV, named +P in [57].

Table 3
Environmental setting. N, population size for each optimization algorithm; D, window size as in LMS algorithm; # of Evaluation is the maximum number of fitness evaluations for each algorithm; Randomized recheck, check the influence of the random number (random initial population) on the performance of an algorithm.

Abbr.	N (Pop size)	Number of Evaluation	D (window size)	Randomized recheck
OFA [43]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
FRCG [44]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
DE [45]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
PSO [46]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
GA [47]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
SHADE [48]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
CMAES [49]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✗
IMODE [50]	10, 13, 15, 17, 20	$N \times 40$	21, 22, ..., 41, 42	✓

the various available algorithms. This study provides statistical evaluations of the performance of the selected algorithms. It does not claim to provide the best algorithms to solve the problem; still only a platform for future researchers to test various other algorithms by integrating them into the framework provided by the authors.

The F1, Se, and PPA are defined as follows:

$$\begin{cases} Se &= \dfrac{TP}{TP+FN} \\ PPA &= \dfrac{TP}{TP+FP} \\ F1 &= 2 \cdot \dfrac{PPA \cdot Se}{PPA+Se} = \dfrac{2 \cdot TP}{2 \cdot TP + FN + FP} \end{cases} \tag{5}$$

We use the QRS detector implemented by the paper [56], simulating the Pan & Tompkins algorithm [59]. The accuracy examination code is based on source Physionet,[5] [60] source code can be found here.[6]

Dataset description The dataset we use is the same as in paper [5], source from Physionet sites [61,62] that holds a freely available database of abdominal and direct fetal electrocardiogram (ADFECGDB).

This database records five pregnant women at birth in their 38th to 41st week of pregnancy. The five-channel recordings were taken at the pulmonary department of the Medical University of Zabrze, Poland.

The individual signals contain:

- A bandwidth of 1 to 150 Hz (with 50 Hz network interference removed).
- A sampling rate of 1 kHz.
- A resolution of 16 bits.
- A length of 5 min.

Electrodes record four aECG signals from the maternal abdomen of an individual, and one is to record the fECG signals directly from the fetus' scalp. To reduce skin impedance, place four silver chloride electrodes around the navel, a reference to the pubic symphysis, and an active electrode on the left lower limb to measure aECG signals. It uses an invasive method - transvaginally by a typical spiral electrode for detecting fECG signals. Through the direct fECG signal measurement, it can label the R-peaks automatically with the sensing system and verify and compile accurate reference annotations by a group of cardiologists. We received five recordings listed on the Physionet site (r01, r04, r07, r08, and r10) and another five recordings (r02, r03, r05, r06, r09) with a sampling rate of 500 Hz from the paper [5].

[5] https://physionet.org/

[6] http://fernandoandreotti.github.io/fecgsyn/pages/examples.html

4.3. Results overview

In the following figures and tables, let D be the window size, a parameter the same as in the LMS algorithm, and N be the population size in an optimization problem.

4.3.1. The overall results

In this section, we summarize the results obtained by each algorithm with their best-performed ones. Results by eight algorithms, CMAES, SHADE, IMODE, GA, OFA, FRCG, PSO, and DE, are shown in Table 4. All algorithms obtained 100% F1, PPA, and Se values with no differences on the r05 recording. Then, in general, eight algorithms perform competitively on r01, r02, r09, and r10, though there are tiny variations in their results according to three F1, PPA, and Se values. The baseline result is 99.4% for r01, 98.8% for r02, 99.2% on r09, and 91.8% on r10 of F1 values. As for recording r08, only FRCG got a 95.7% F1 value, which is lower than others. The other seven algorithms got 99.4% and 99.5% with 0.1% up and down values on the F1 value. Performance on recording r03 and r06 are quite different by tested algorithms. IMODE got the best result on r03 with 98.5% F1 value, GA, and FRCG got over 98% F1 value; DE, PSO, and SHADE got over 97% F1 value, while CMAES got the worst result with 91.4% F1 value and OFA got the second last result of 96.3% on r03 recording. CMAES got the best result on recording r06, and SHADE got the second-best result; all other algorithms got competitive F1 values around 94%. It is quite hard to get high accuracy on recording r04 and r07, where PSO performed the best at 84.6% F1 value and 84.2% F1 value by FRCG on r04 and r07, respectively. With limited window side varied space, eight algorithms perform competitively on six out of 10 instances. As a comparison to the baseline, we put the results of LMS in Appendix.

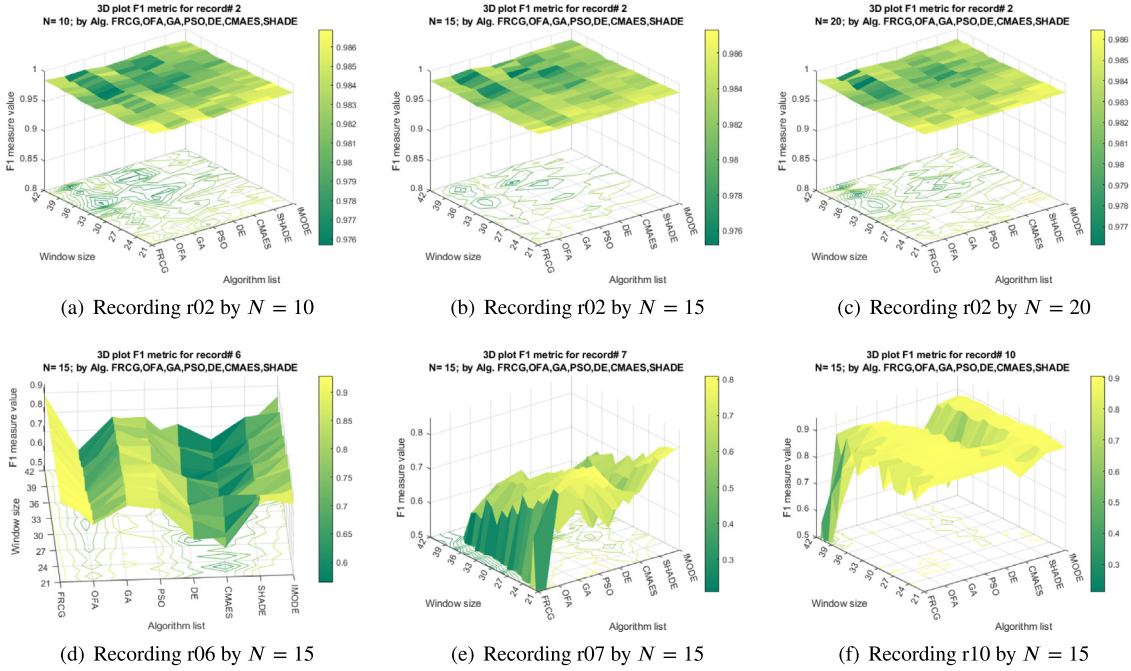
4.3.2. Result comparison among individual algorithms

The result from Table 4 shows no significant difference among each MA; they perform competitively on most data. However, Table 4 result shows the best-performed ones under all possible experimental settings, such as various window sizes. That is to say, the performance of each MA may produce inconsistent results under one parameter setting. The experiment in this section is to answer the question, 'if the parameter (window size) influences the performance of signal extraction heavily?'.

The experiment tests the performance of each MA with a varied window size parameter and gets the average accuracy value under 22 independent runs. We set the window size from 21 to 42. Fig. 5 plots a 3D image of obtained results on a single data (recording) by all compared algorithms. Fig. 5(a) shows the F1 accuracy results on recording r02 with a population size of 10. The color bar beside the sub-figures of Fig. 5 shows the range of the result values. All algorithms' minimum and maximum F1 values are between 0.97 and 0.988, corresponding to the darkest to the light color. Though the color changes big in the 3D image Fig. 5(a), actually, the values are less than 0.015 difference for all.

Table 4
The best-performed F1, PPA, and Se accuracy result by MAs (CMAES, SHADE, IMODE, GA, OFA, FRCG, PSO, and DE) through all the experiments on ten data r01 to r10.

Recording	CMAES			SHADE			IMODE			GA			OFA			FRCG			PSO			DE		
	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se	F1	PPA	Se
r01	99.5	99.4	99.5	99.4	99.4	99.5	99.5	99.4	99.5	99.5	99.4	99.5	99.5	99.4	99.5	99.1	99.1	99.2	99.5	99.4	99.5	99.5	99.4	99.5
r02	98.9	98.8	99.1	98.9	98.8	99.1	98.8	98.6	98.9	98.8	98.6	98.9	98.8	98.5	99.1	98.8	98.6	98.9	98.8	98.6	98.9	99.1	98.9	99.2
r03	91.4	92.4	90.4	97.2	96.9	97.4	**98.5**	98.4	98.5	98.3	98.1	98.5	96.3	96.2	96.3	98.2	98.2	98.2	97.2	97.2	97.2	97.9	98.0	98.0
r04	77.4	76.6	78.2	82	81.8	82.1	80.2	79.9	80.7	82.1	81.7	82.5	79.2	78.6	79.8	69.9	69.2	70.6	**84.6**	84.7	84.5	80.5	80.1	81.4
r05	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
r06	**96.5**	96.9	96.1	95.5	95.5	95.4	94.2	94.1	94.7	93.8	93.5	94.1	94.6	94.5	94.8	94.0	93.9	94.2	94.1	93.7	94.5	94.9	95.2	94.5
r07	79.3	78.5	80.2	81.1	80.4	81.9	82.2	81.6	82.9	81.8	81.0	82.6	81.2	80.6	82.1	**84.2**	83.0	85.4	83.6	82.8	84.5	83.1	82.5	83.7
r08	99.4	99.2	99.7	99.5	99.4	99.7	99.5	99.4	99.7	99.5	99.4	99.7	99.5	99.4	99.7	95.7	95.3	96.2	99.5	99.4	99.7	99.5	99.4	99.7
r09	99.2	99.2	99.2	99.3	99.2	99.4	99.3	99.2	99.4	99.2	99.1	99.4	98.8	98.6	99.1	99.4	99.2	99.5	99.0	98.8	99.2	99.1	98.9	99.2
r10	92.3	90.0	94.8	92.3	90.0	94.8	91.9	90.1	94.1	91.9	89.6	94.3	92.8	90.5	95.2	92.1	89.9	94.4	92.0	89.5	94.6	91.8	89.4	94.3



(a) Recording r02 by $N = 10$ (b) Recording r02 by $N = 15$ (c) Recording r02 by $N = 20$

(d) Recording r06 by $N = 15$ (e) Recording r07 by $N = 15$ (f) Recording r10 by $N = 15$

Fig. 5. 3D plot of average results of 10 runs on a single recording by optimization algorithms CMAES, SHADE, IMODE, GA, OFA, FRCG, PSO and DE with the window size range from 21 to 42.

Fig. 5(b) shows the results by population size of 15, and the F1 value changes for all are in less than 0.01, 0.01 for results with $N = 20$ as in Fig. 5(c). Hence, we can conclude that the window size in a specific range has a trivial influence on the final F1 value on the r02 recording. Fig. 5(d) show that OFA and CMAES did not get F1 values consistently as the 2D contour has a gradient *green* color, which blue curves represent the worst values compared to the values around. PSO and DE also show bad results on the window size around 28 and 23 around, respectively. For Fig. 5(e) of r07 recording, except for FRCG and CMAES, other algorithms show few vibrations in their results only with a window size of 21 around. More importantly, other algorithms perform poorly with window sizes larger than 36. In Fig. 5(f), it is obvious that FRCG and CMAES perform unstable with varied window sizes; all other algorithms produce approximative F1 values under varied window size conditions.

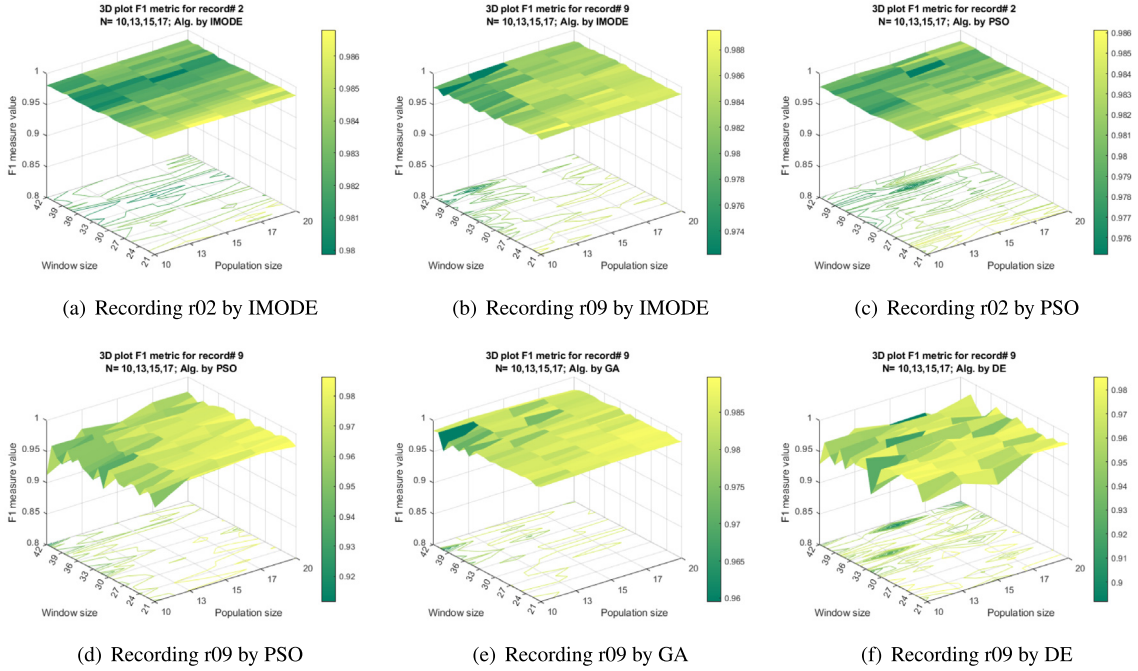
4.3.3. Performance on different population size

This section provides the performance under different population sizes, ranging from 10 to 20. We set a small population and a low allowed fitness evaluation number to keep the computation cost low.

Fig. 6 shows the results by IMODE, PSO, and DE experimented on various population sizes. Fig. 6(a)(b),(c)(d) shows the result on recording r02, r09 by IMODE and PSO, respectively. The 3D plot of the r02 result has small waves by IMODE. However, the values are between 0.986 to 0.98, with 0.006 fluctuating up and down. For r09, the values are between 0.988 to 0.974, with 0.014 up and down. PSO performs consistently well on r02 recording with F1 values up to 0.975; the highest F1 value reaches 0.986 with less than 0.01 changes on all tested population sizes. The F1 value changes big on recording r09 with 0.1 up and down by PSO. GA performs badly with a population size of 25 and a window size of 39, reaching the lowest F1 value of around 0.96. GA obtained as high as 0.98 F1 value with less than 0.02 up and down changes on recording r09. There are three cases where DE performs badly, a population size of 17 and a population size of 13 and 10, with a window size of 32, 35, and 40, respectively. For other cases on r09, DE reaches 0.98 F1 value and 0.1 around up and down changes. IMODE performs consistently better with various population sizes and window sizes than other compared algorithms. From Fig. 6, we conclude that a small population size difference would not significantly influence the results concerning specific recordings.

4.3.4. Statistical comparisons

Statistical comparisons are practical and essential to evaluate the algorithm's performance, leading to reliable conclusions. This



(a) Recording r02 by IMODE (b) Recording r09 by IMODE (c) Recording r02 by PSO

(d) Recording r09 by PSO (e) Recording r09 by GA (f) Recording r09 by DE

Fig. 6. 3D plot of average results of 10 runs on a single recording by optimization algorithms IMODE, GA, PSO, and DE. The window size ranges from 21 to 42 and the population size is set to 10, 13, 15, 17, and 20.

section delivers a statistical comparison of all tested algorithms. The aim of this comparison is to show which algorithm performs better (though slightly) than others, which might be findings for future investigation.

We adopt the bayesian-based frequentist statistics instead of the Null Hypothesis Statistical Test (NHST) [63] for the analysis of the results. The bayesian method executes a distribution of the parameter of interest rather than a probability value calculation, reducing the burden of no significant comparison by NHST. The bayesian-based test starts with a mathematical model of the obtained data and then selects a prior distribution. At last, it uses Bayes' rule to infer the posterior distribution of the samples from data [64]. Finally, we perform Bayesian Signed-Rank tests on the results of optimization algorithms by comparing the average accuracy value to analyze which algorithm performs better on average. In this test, three values θ_l, θ_e and θ_r are computed, which denotes the probabilities of the mean difference being in the intervals $(-\infty, -r)$, $[-r, r]$ or $(r, +\infty)$ between two algorithms respectively [64]. Additionally, we need to set the region of practical equivalence (abbr. as *rope*) during the Bayesian test procedure, which is also the parameter r. Readers can find more details in [65].

Fig. 8 shows the results of the statistical comparison. First, the comparison data of each algorithm contains ten values representing ten recording signal F1 accuracy, and each value is an average result of all obtained 22 values under 22 different window size parameters. We set the r (*rope*) to 0.05 for a rough comparison. In Fig. 8, '*blue*' (light *blue* to dark *blue*) colored squares denote the compared two algorithms (from Algorithm 1 of the X-axis and Algorithm 2 of the Y-axis) is a tie with a greater probability. Specifically, as inferred in the 'Winner' bar, the '*blue*' color marks the '*rope*' as the winner in the competition between the two algorithms. Meanwhile, the shade of the color represents the degree of probability value; the darker the color is, the more significant probability it gets. For example, the '*green*' colored square shows the 'winner' is from the 'Alg.2' side, which means the Y-axis algorithm performs much better than the algorithm from the X-axis if the color of the square is a dark green. The same is applied to the '*orange*' color, denoting an algorithm's winner from the X-axis.

Fig. 8(a)–(c) shows the statistical F1 values on results with population size ten, fifteen, and twenty. The '*blue*' squares present algorithms DE, GA, IMODE, PSO, and SHADE are tied with high probability when we set the r equal to 0.05. The PPA results as shown in Fig. 8(d)–(f) also imply that algorithms DE, GA, IMODE, PSO, and SHADE equally belong to the first classified algorithms, and CMAES, FRECG perform worse compared to others. Besides, OFA is tied to the first classified algorithms but with a low probability, showing a light *blue* color corresponding to the OFA algorithm.

In general, IMODE and PSO perform slightly better than others from that Table 4 results. To further investigate which algorithm is superior to others, we separately conduct a paired comparison and set the $r = 0.01$. The results are shown in Fig. 6, where IMODE wins at the F1 value with a population size of 10 compared to PSO as in Fig. 6(a). Furthermore, IMODE and PSO are a tie with high probability, as in Fig. 6(b)(c) with experiments with a population size of fifteen and twenty.

To summarize, IMODE performs better than other selected algorithms from the results above. One of the reasons may lie in the multi-selection of updating operators. The IMODE algorithm directs the population for exploration based on the best updating rules in options and is not limited to a specific searching style. This multi-operator searching is good for multi-model and composite problems. The second average best algorithms are PSO and GA, with relatively low sensitivity by window size and population size. The possible reason for this low sensitivity may be from the simple updating equations.

4.3.5. Performance with/without amplitude tuning (AT)

This section verifies the effectiveness and necessity of operating the AT after getting the signal fECG* obtained by optimization algorithms. AT operation is to repair the amplitude of the acquired signal by tuning the noise. It only goes through the data once and costs $\mathcal{O}(n)$. At the same time, it is easier and cheap than WT operation in paper [5] or other decomposition-based transformations.

Table 5
Comparison results of F1 value between signals obtained by IMODE with/without AT operation, where 'BlueGreen' marks the results without AT operation, 'Cerulean' denotes the result with AT; and 'red' colors the F1 improvements by AT than those without AT and 'blue' colors the F1 value decrease accordingly.

W	Result without AT, R_b										Result with AT, R_a										Results difference $df = R_a - R_b$									
	r01	r02	r03	r4	r05	r06	r07	r08	r09	r10	r01	r02	r03	r4	r05	r06	r07	r08	r09	r10	r01 → r10									
$N = 10$																														
D21	99.2	98.6	96.9	33.8	100	66.0	37.9	99.1	98.9	36.5	99.2	98.6	96.9	69.2	100	91.4	78.2	99.1	98.9	88.0	0	0	0	35.3	0	25.5	40.3	0	0	51.5
D22	98.6	98.6	92.6	34.5	100	93.0	37.5	99.3	97.3	37.4	98.6	98.6	92.6	80.2	100	93.0	76.4	99.3	97.3	89.8	0	0	0	45.7	0	0	39.0	0	0	52.4
D23	99.1	98.6	88.4	31.2	99.8	89.3	37.7	99.2	98.7	38.0	99.1	98.6	88.4	71.8	99.8	93.5	78.3	99.2	98.7	89.4	0	0	0	40.6	0	4.2	40.7	0	0	51.4
D24	98.3	98.6	87.1	31.4	100	90.9	35.4	99.2	98.2	37.3	98.3	98.6	87.1	77.7	100	90.9	77.2	99.2	98.2	89.9	0	0	0	46.3	0	0	41.7	0	0	52.6
D25	98.8	98.5	90.7	30.6	100	91.1	36.6	99.0	98.5	37.0	98.8	98.5	90.7	67.4	100	91.1	75.2	99.0	98.5	90.2	0	0	0	36.8	0	0	38.6	0	0	53.2
D26	99.0	98.6	95.0	31.3	100	92.6	33.8	98.9	98.4	36.8	99.0	98.6	95.0	74.6	100	92.6	65.8	98.9	98.4	89.6	0	0	0	43.3	0	0	31.9	0	0	52.8
D27	98.3	98.4	96.5	63.3	99.5	93.8	37.5	99.2	98.2	89.0	98.3	98.4	96.5	63.7	99.5	93.8	74.0	99.2	98.2	89.0	0	0	0	0.3	0	0	36.5	0	0	0
D28	98.3	98.5	93.7	33.8	100	91.8	64.9	99.2	98.0	89.9	98.3	98.5	93.7	70.7	100	91.8	65.1	99.2	98.0	89.9	0	0	0	36.8	0	0	0.2	0	0	0
D29	99.2	98.6	94.0	63.3	99.8	88.5	72.4	99.2	98.4	34.1	99.2	98.6	94.0	63.4	99.8	88.6	72.6	99.2	98.4	91.2	0	0	0	0.2	0	0.1	0.2	0	0	57.1
D30	99.2	98.4	90.5	76.1	99.9	78.6	53.5	99.4	98.5	91.9	99.2	98.4	90.5	76.6	99.9	88.6	53.4	99.4	98.5	91.9	0	0	0	0.4	0	10.0	−0.1	0	0	0
D31	98.8	98.5	84.9	73.8	99.6	89.3	68.9	99.1	99.1	89.3	98.8	98.5	84.9	74.1	99.6	89.3	68.9	99.1	99.1	89.3	0	0	0	0.2	0	0	0	0	0	0
D32	98.5	97.7	84.6	72.2	99.9	90.3	65.5	99.3	97.8	24.1	98.5	97.7	84.6	72.6	99.9	90.3	65.6	99.3	97.8	90.2	0	0	0	0.4	0	0	0.1	0	0	66.1
D33	99.3	98.5	73.3	72.6	99.9	73.7	66.0	99.2	98.2	89.3	99.3	98.5	73.3	73.4	99.9	73.7	65.8	99.2	98.2	89.3	0	0	0	0.8	0	0	−0.2	0	0	0
D34	98.9	98.0	66.3	75.4	100	74.3	58.8	99.3	97.5	88.6	98.9	98.0	66.3	76.2	99.9	74.3	59.0	99.3	97.5	88.6	0	0	0	0.8	−0.1	0	0.2	0	0	0
D35	98.9	97.7	68.2	74.4	99.9	56.7	57.8	99.2	98.0	88.6	98.9	97.7	68.2	74.6	99.9	82.7	57.9	99.2	98.0	88.6	0	0	0	0.3	0	26.0	0.1	0	0	0
D36	99.2	97.7	84.4	78.1	99.6	76.2	63.7	99.0	98.0	85.5	99.2	97.7	84.4	78.7	99.6	88.2	63.8	99.0	98.0	85.5	0	0	0.1	0.5	0	12.0	0.1	0	0	0
D37	99.5	98.2	85.8	69.7	99.9	91.4	59.9	99.1	97.8	89.1	99.5	98.2	85.8	70.1	99.9	91.4	60.1	99.1	97.9	89.1	0	0	0	0.4	0	0	0.2	0	0.1	0
D38	99.1	98.5	69.0	57.3	99.5	91.1	68.0	99.5	98.2	90.6	99.1	98.5	69.0	57.8	99.5	91.1	68.2	99.5	98.2	90.6	0	0	0	0.6	0	0	0.2	0	0	0
D39	99.3	97.9	71.4	57.6	99.8	78.2	38.2	99.1	97.7	90.3	99.3	97.9	71.4	57.9	99.8	78.2	38.1	99.1	97.7	90.3	0	0	0	0.4	0	0	−0.1	0	0	0
D40	99.1	98.2	60.6	48.4	99.8	88.0	54.8	99.3	97.7	89.3	99.1	98.2	60.6	48.5	99.8	88.0	54.9	99.3	97.7	89.3	0	0	0	0.1	0	0	0.1	0	0	0
D41	99.2	98.6	62.1	38.0	99.8	63.6	44.5	99.2	97.6	89.8	99.2	98.6	62.1	38.2	99.8	63.6	44.4	99.2	97.6	89.8	0	0	0	0.2	0	0	−0.1	0	0	0
D42	98.8	98.0	79.9	33.2	99.3	80.8	22.0	99.3	97.6	88.1	98.8	98.0	79.9	33.4	99.3	80.8	22.1	99.3	97.6	88.1	0	0	0	0.2	0	0	0.1	0	0	0
$N = 20$																														
D21	99.1	98.2	90.5	25.0	100	19.0	22.9	99.2	98.1	88.0	99.1	98.2	90.5	25.4	100	90.4	23.0	99.2	98.1	88.0	0	0	0	0.4	0	71.4	0	0	0	0
D22	98.8	98.8	94.1	32.9	100	91.8	37.2	99.1	99.3	37.2	98.8	98.8	94.1	75.7	100	91.8	79.6	99.1	99.3	88.8	0	0	0	42.8	0	0	42.4	0	0	51.6
D23	99.1	98.4	96.0	33.4	100	93.2	37.1	98.9	98.8	37.6	99.1	98.4	96.0	78.4	100	93.2	80.3	98.9	98.8	89.0	0	0	0	44.9	0	0	43.2	0	0	51.4
D24	98.5	98.4	96.1	33.3	100	92.5	37.3	99.3	99.3	37.5	98.5	98.4	96.1	80.1	100	92.5	79.4	99.3	99.3	90.0	0	0	0	46.8	0	0	42.1	0	0	52.5
D25	99.1	98.3	85.5	30.4	100	93.0	35.3	99.3	98.8	37.9	99.1	98.3	85.5	71.0	100	93.0	72.7	99.3	98.8	90.7	0	0	0	40.7	0	0	37.4	0	0	52.8
D26	98.0	98.3	90.7	31.6	99.9	94.2	36.1	99.3	99.3	36.8	98.0	98.3	90.7	75.9	99.9	94.2	71.7	99.3	99.3	90.0	0	0	0	44.3	0	0	35.6	0	0	53.3
D27	98.4	98.1	86.8	30.0	99.8	89.6	36.8	99.3	98.6	39.0	98.4	98.1	86.8	71.3	99.8	89.6	72.3	99.3	98.6	90.3	0	0	0	41.3	0	0	35.6	0	0	51.3
D28	99.0	98.2	97.1	31.3	99.5	93.2	36.2	99.2	98.3	90.2	99.0	98.2	97.1	74.7	99.5	93.2	75.5	99.2	98.3	90.2	0	0	0	43.4	0	0	39.3	0	0	0
D29	98.4	98.5	88.6	32.8	100	90.1	35.5	99.2	99.0	90.3	98.4	98.5	88.6	74.0	100	90.1	73.2	99.2	99.0	90.3	0	0	0	41.2	0	0	37.7	0	0	0
D30	99.0	98.2	84.2	70.3	100	91.3	71.7	99.4	98.6	36.2	99.0	98.2	84.2	70.9	100	91.3	71.8	99.4	98.6	88.7	0	0	0	0.6	0	0	0.1	0	0	52.5
D31	98.7	98.0	90.5	35.1	100	90.9	39.4	99.1	98.3	89.5	98.7	98.0	90.5	78.9	100	90.9	71.7	99.1	98.3	89.5	0	0	0	43.8	0	0	32.3	0	0	0
D32	99.2	98.5	89.7	74.1	100	90.4	61.0	99.2	99.1	89.6	99.2	98.5	89.7	74.8	100	90.4	61.0	99.2	99.1	89.6	0	0	0	0.7	0	0	0	0	0	0
D33	98.2	98.0	90.8	73.0	100	91.5	67.8	98.9	98.2	91.4	98.2	98.0	90.8	73.4	100	91.5	67.8	98.9	98.2	91.4	0	0	0	0.4	0	0	0.1	0	0	0
D34	99.1	98.5	83.1	67.5	99.9	92.0	60.8	99.2	98.6	87.7	99.1	98.5	83.1	68.3	99.9	92.0	61.4	99.2	98.6	87.7	0	0	0	0.8	0	0	0.6	0	0	0
D35	99.1	98.2	89.8	77.4	100	92.1	73.1	99.3	98.3	88.9	99.1	98.2	89.8	78.1	100	92.1	73.6	99.3	98.3	88.9	0	0	0	0.7	0	0	0.4	0	0	0
D36	99.1	98.1	84.0	73.3	99.9	89.6	68.8	99.2	98.6	88.8	99.1	98.1	84.0	74.1	99.9	89.6	69.0	99.2	98.6	88.8	0	0	0	0.8	0	0	0.2	0	−0.1	0
D37	99.3	98.5	96.7	68.6	100	84.7	69.9	99.2	98.3	90.2	99.3	98.5	96.7	69.0	100	84.8	70.4	99.2	98.3	90.2	0	0	0	0.4	0	0.1	0.5	0	0	0
D38	99.1	98.3	80.3	60.8	100	88.7	62.7	99.2	98.9	89.5	99.1	98.3	80.3	61.0	100	88.7	62.9	99.2	98.9	89.5	0	0	0	0.2	0	0	0.2	0	0	0
D39	99.2	98.3	97.5	51.8	100	92.3	50.1	99.2	97.7	89.9	99.2	98.3	97.5	52.0	100	92.3	50.2	99.2	97.7	89.9	0	0	0	0.2	0	0	0.1	0	0	0
D40	99.3	98.6	82.5	55.0	99.8	90.3	35.6	99.2	97.5	90.1	99.3	98.6	82.5	55.3	99.8	90.3	35.6	99.2	97.5	90.1	0	0	0	0.3	0	0	0	0	0	0
D41	98.8	98.0	65.8	44.4	99.8	86.1	28.8	99.3	98.5	89.4	98.8	98.0	65.8	44.9	99.8	86.1	28.8	99.3	98.5	89.4	0	0	0	0.5	0	0	0	0	0	0
D42	98.9	98.0	68.1	33.8	99.8	26.9	35.0	99.1	99.1	86.4	98.9	98.0	68.1	33.9	99.8	90.8	35.2	99.1	99.1	86.4	0	0	0	0.1	0	63.9	0.2	0	0	0

Due to the page limit, we only show the comparison results with and without the AT operation on the filter assisted by IMODE and PSO. They perform slightly consistently better on all recordings. The experimental result is shown in Table 5 by IMODE, where R_b denotes the F1 value results on the signal obtained directly from IMODE without AT operation. In comparison, R_a denotes the F1 value in the AT-operated fECG result after the IMODE process. $df = R_a - R_b$ indicates the improvement by AT operation to the results without AT function.

Ten recordings results of df are in the last block of Table 5. The positive value in the df result denotes the improvement by AT operation. The negative value means the impact worsens after operating the AT, and zero indicates no changes in the F1 value between signals using AT or without using AT operation.

Table 5 show the results with a population size equal to 10 by IMODE. As columns *result without AT* of R_b shows, IMODE gets 36% approximately F1 value on recording r10 when the window size D are 21 to 25. That F1 values are much lower than the values in D of 33 to 42 with 89% F1 value. These changes on the window side influence the F1 value largely. However, from columns *result with AT* of R_a, it shows F1 values of recording r10 are all above 85%. We can see F1 value improvement from df, where r10 results are up to 50% at the window size of 21 to 25 and window size of 29 and 32. A similar improved F1 value also appears in the results in a population size of 20. The F1 value reaches higher on recording r04, r07, and r10 with 35% approximately on results in varied window sizes.

Table 6 shows the improvement in F1 value with AT operation on the results obtained by the PSO algorithm. The results show a population size of 10, 15, and 20. When the results are good (e.g., F1 value is above 60%), the operation of AT is of no use for the signal F1 value improvement, getting the same result as the one without AT. However, once the result R_b is terrible, bringing 30%–60% around of F1 value, the advance by AT operation is noticeable. Though there are eleven spotted negative values in Table 6, the results worsen than those without AT operation. However, most poor values are 0.1%, except for one value is 0.4%. Compared to the improvement of largely 50% up on the F1 value, we can ignore changes 0.1 up and down. Generally, the AT operation is effective and practical for improving signal extraction.

4.3.6. Results with different random seed

In this section, we use Autorank [66] to test the randomness influence on results by experimenting with the same task multiple runs. Autorank is an analysis tool for non-experts, which

Table 6
Comparison results of F1 value between signals obtained by PSO with/without AT operation, Abbr. refer to Sec. where 'red' colors the F1 improvements by AT than those without AT and 'blue' colors the F1 value decrease correspondingly.

W	r01 → r10, $N = 10$, $df = R_a - R_b$									r01 → r10, $N = 15$, $df = R_a - R_b$									r01 → r10, $N = 20$, $df = R_a - R_b$								
D21	0	0	8.3	38.5	0	0	40.3	0	0	51.9	0	0	−0.1	44.4	0	0	45.0	0	0	52.8	0	0	0	45.5	0	0	43.8	0	0	52.1
D22	0	0	0	43.0	0	0	39.9	0	0	53.0	0	0	−0.1	44.6	0	0.1	43.8	0	0	53.4	0	0	0	46.0	0	0	46.0	0	0	52.8
D23	0	0	37.2	44.3	0	0	38.5	0	0	52.7	0	0	0	41.0	0	3	40.9	0	0	0	0	0	0	46.3	0	0	41.8	0	0	52.5
D24	0	0	0	42.7	0	4.9	46.5	0	0	53.9	0	0	0	44.9	0	0	42.2	0	0	52.7	0	0	0	40.0	0	0	43.9	0	0	53.0
D25	0	0	0	40.8	0	32.9	42.8	0	0	51.4	0	0	0	38.6	0	0	42.4	0	0	50.3	0	0	0	44.1	0	0	37.7	0	0	0
D26	0	0	18.9	41.8	0	35.9	39.8	0	0	51.9	0	0	0	43.8	0	0	43.1	0	0	0	0	0	0	43.6	0	0	43.7	0	0.1	52.2
D27	0	0	0	33.1	0	0	42.0	0	0	52.4	0	0	0	42.8	0	0.1	44.3	0	0	0	0	0	0	42.4	0	0	0	0	0	0
D28	0	0	0	0.2	0	0	37.9	0	0	52.1	0	0	0	40.8	0	0	0.1	0	0	53.4	0	0	0.1	38.9	0	0	35.0	0	0	53.0
D29	0	0	0	0.1	0	0	0.1	0	0	56.8	0	0	0	0.4	0	9.4	0.1	0	0	0	0	0	0	0.2	0	0	0	0	0	0
D30	0	0	0	0.1	0	0	36.6	0	0	0	0	0	14.2	45.9	0	26.7	0	0	0.2	0	0	0	−0.1	0.5	0	0	0.2	0	0	0
D31	0	0	0	0.2	0	0	38.2	0	0	0	0	0	0	0.5	0	0	0.2	0	0	0	0	0	0	0	0	7.9	26.9	0	0	0
D32	0	0	0	0.3	0	0	0.1	0	0.1	0	0	0	4.9	0.7	0	26.4	0.2	0	0	0	0	0	0	0.5	0	0	0	0	0	0
D33	0	0	0	38.0	0	30.1	0.2	0	0	0	0	0	0	0.4	0	−0.1	0.4	0	0	0	0	0	0	0.3	0	0	0.3	0	0	0
D34	0	0	0	0.8	0	6.5	28.8	0	0	0	0	0	32.9	0.5	0	0	0.2	0	0	0	0	0	0	0.5	0	0	−0.1	0	0	0
D35	0	0	0	0.9	0	20.6	27.8	0	0	75.0	0	0	0	0.8	0	22.0	0.3	0	0	0	0	0	0	0.4	0	0.1	0.3	0	0	0
D36	0	0	0	0.7	0	0	0	0	0	0	0	0	0	0.5	0	0	0.1	0	0	0	0	0	0	0.7	0	0	0.3	0	0	0
D37	0	0	0	0.6	0	4.3	−0.1	0	0	0	0	0	0	0.6	0	28.7	0.3	0	0	0	0	0	0	0.6	0	0.1	0.2	0	0	0
D38	0	0	0	0.2	0	0	0.1	0	0	0	0	0	0	0.4	0	0	0.2	0	0	0	0	0	0	0.4	0	23.5	0.1	0	0	0
D39	0	0	−0.1	0	0	29.1	0.2	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	−0.1	0.4	0	−0.1	0.2	0	0	0
D40	0	0	0.1	0.4	0	0	0.1	0	0	75.1	0	0	0	0.4	0	0	0.3	0	0	0	0	0	0	0.3	0	0	0	0	0	0
D41	0	0	0	−0.1	0	0	0.4	0	0	0	0	0	0	0.4	0	0	0.1	0	0.1	0	0	0	0	0.5	0	0	0.2	0	0	0
D42	0	0	0	−0.4	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0	−0.1	0	0	0	0	0.4	0	0	0.1	0	0	0

Table 7
Summary of random seed influence on IMODE algorithm on 14 runs on Recordings r01 to r05, each result labeled as IMODE(run_ID), where *ID* denotes a number in [0,1, . . . , 13]; *M, SD* denotes the mean, the Std. value of F1 result, respectively; *CI* denotes the confidence interval, which the upper value is up to 1.

	M	*SD*	CI	magnitude
IMODE(run_4)	0.880	0.155	[0.718, 1.042]	negligible
IMODE(run_9)	0.884	0.152	[0.722, 1.046]	negligible
IMODE(run_6)	0.879	0.159	[0.717, 1.041]	negligible
IMODE(run_1)	0.892	0.145	[0.730, 1.053]	negligible
IMODE(run_8)	0.897	0.144	[0.735, 1.059]	negligible
IMODE(run_0)	0.889	0.142	[0.728, 1.051]	negligible
IMODE(run_12)	0.890	0.146	[0.728, 1.051]	negligible
IMODE(run_7)	0.888	0.146	[0.726, 1.050]	negligible
IMODE(run_2)	0.897	0.138	[0.735, 1.058]	negligible
IMODE(run_13)	0.894	0.145	[0.732, 1.055]	negligible
IMODE(run_11)	0.888	0.147	[0.726, 1.050]	negligible
IMODE(run_10)	0.895	0.139	[0.733, 1.057]	negligible
IMODE(run_5)	0.896	0.142	[0.734, 1.057]	negligible
IMODE(run_3)	0.895	0.143	[0.733, 1.057]	negligible

Table 8
Summary of random seed influence on IMODE algorithm on 14 runs on Recordings r01 to r05, each result labeled as IMODE(run_ID), where *ID* denotes a number in [0,1, . . . , 13]; *M, SD* denotes the mean, Std. value of PPA result, respectively.

	M	*SD*	CI	magnitude
IMODE(run_9)	0.882	0.154	[0.717, 1.046]	negligible
IMODE(run_6)	0.877	0.162	[0.713, 1.042]	negligible
IMODE(run_4)	0.879	0.156	[0.714, 1.043]	negligible
IMODE(run_1)	0.890	0.147	[0.725, 1.054]	negligible
IMODE(run_8)	0.895	0.147	[0.731, 1.059]	negligible
IMODE(run_0)	0.887	0.145	[0.723, 1.052]	negligible
IMODE(run_13)	0.891	0.148	[0.727, 1.056]	negligible
IMODE(run_12)	0.888	0.148	[0.723, 1.052]	negligible
IMODE(run_2)	0.895	0.140	[0.730, 1.059]	negligible
IMODE(run_7)	0.887	0.148	[0.722, 1.051]	negligible
IMODE(run_5)	0.893	0.145	[0.729, 1.058]	negligible
IMODE(run_11)	0.886	0.149	[0.722, 1.051]	negligible
IMODE(run_10)	0.893	0.141	[0.729, 1.058]	negligible
IMODE(run_3)	0.893	0.146	[0.729, 1.058]	negligible

requires no knowledge from the users to decide the specific statistical method for their data. Instead, Autorank chooses the statistical test method based on the data in a decision-flow way and provides interpretation results for the user.

We tested on IMODE algorithm on recording r01 to r05 in 14 runs with randomly generated seed. For each recording, we use a window size ranging from 21 to 42 and get 22 instances of results (F1, PPA). Then we average those 22 results as a final value concerning that recording. Thus, table data with five rows (recordings) and 14 columns (different runs) is our test data. For each run we mark with IMODE(*run_id*), where $id = [0, 1, 2, \ldots, 13]$.

Tables 7 and 8 show summary results on the IMODE algorithm in 14 runs with different random seeds for F1, and PPA, respectively. The magnitude denotes the degree of difference between algorithms. Suppose the mean rank difference is greater than the critical distance (CD) of the Nemenyi test. The magnitude of Differences between examples is significant. In our cases, the difference is negligible among all random runs, where the CD is None. From the table's results, we can conclude that the randomness has negligible effects on the final results.

The results of the mean value in Tables 7 and 8 are lower compared to the result in Table 4. As Table 4 shows the mean value of one recording, while the values in Tables 7 and 8 are the mean values of five recordings.

The fECG extraction model adopts eight optional MAs to generate the coefficient (signal weights). From the results and figures above, we observe that POAs can extract effective coefficients compared to the classical LMS methods (results of LMS are shown in Table A.1. Additionally, we provide Table A.2 showing the state-of-the-art methods' results where we use the same recordings.) on all recordings used in this work. However, for recording, such as $r01, r05$, the optimization algorithms can obtain consistent results regardless of the population size and window size changes. But POAs failed on recording such as $r04, r07$ as LMS did. This result difference leads to the thought that the quality of extracted *aECG* and *mECG* affects the performance of POA. Tables 5 and 6 confirm that the AT process can further improve the POA performance, reducing the influence of hyperparameters. In general, the FRCG algorithm performs worse than other algorithms, which implies that the fECG extraction problem may be a multi-modal problem with complex local minima. IMODE algorithm operates the best-oriented updating operators, which helps avoid the local minima as IMODE performs the best.

Table 9
F1 and PPV accuracy results of CEC2022EA4eig on ten recordings (r01-r10) with a window size of D20 to D42. The population size is set to 10.

F1	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42
r01	98.8	98.4	99.0	98.9	98.9	99.0	98.6	98.3	98.8	98.9	98.9	98.5	98.4	98.3	97.3	98.2	98.5	98.6	98.2	98.7	98.1	98.6	98.5
r02	98.6	98.6	98.7	98.5	98.5	98.7	98.4	98.3	98.3	98.3	98.3	98.2	98.2	98.2	98.2	98.1	97.9	98.2	98.1	98.2	98.2	98.2	98.2
r03	80.6	84.3	81.3	84.2	77.8	77.5	83.2	90.4	75.2	80.9	72.4	67.6	66.8	63.6	61.9	68.6	62.4	58.7	69.2	63.6	59.5	56.7	68.9
r04	78.2	78.0	73.4	72.2	73.6	70.2	68.6	67.4	66.4	71.5	66.1	70.5	71.5	59.7	71.0	66.8	60.9	64.8	63.3	53.1	47.9	36.0	34.6
r05	99.8	99.7	99.5	99.8	99.9	99.9	99.7	99.8	99.7	99.7	99.7	99.5	99.8	99.4	99.3	99.6	99.7	99.1	99.1	99.4	99.0	99.5	99.2
r06	88.6	85.6	83.7	86.1	81.6	84.0	87.4	85.6	86.9	84.4	75.0	78.9	77.6	83.9	77.9	75.6	76.2	76.4	67.6	71.8	73.2	69.9	72.0
r07	79.7	78.8	78.1	78.4	77.0	74.0	72.2	71.4	69.1	68.3	69.4	71.4	61.1	60.2	70.6	71.6	66.7	61.3	56.1	45.5	42.2	45.6	28.9
r08	99.2	99.1	99.2	99.1	99.1	99.1	99.0	99.1	99.2	99.1	99.0	99.2	99.1	99.0	99.2	99.1	99.3	99.1	99.0	99.2	99.1	98.8	99.0
r09	98.5	98.5	98.1	98.3	98.4	98.3	98.0	98.2	98.2	98.2	98.1	97.5	97.6	98.0	98.1	97.4	95.3	96.8	97.8	97.7	97.3	97.3	95.2
r10	88.8	88.6	89.0	89.6	89.5	89.7	90.6	90.3	90.5	90.3	89.8	89.5	89.2	88.4	89.0	88.7	88.2	89.1	89.6	89.8	88.9	88.1	87.6
PPV	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42
r01	98.6	98.4	99.0	98.8	98.8	98.9	98.5	98.2	98.7	98.9	98.9	98.4	98.3	98.2	97.1	98.0	98.4	98.5	97.9	98.6	98.0	98.5	98.4
r02	98.4	98.4	98.5	98.3	98.2	98.5	98.1	98.0	98.0	98.0	98.1	98.0	98.0	97.9	97.9	97.8	97.7	97.9	97.8	97.9	97.9	98.0	97.8
r03	81.6	84.7	81.7	84.5	77.8	78.5	83.6	90.8	75.4	81.3	74.1	68.3	66.9	64.8	62.4	70.4	63.2	61.8	71.2	63.9	60.8	60.1	69.5
r04	78.0	77.4	72.6	71.0	72.6	69.3	67.4	66.1	65.0	70.3	65.0	69.0	70.5	58.6	70.1	65.8	59.6	63.5	62.3	52.1	47.0	35.2	33.9
r05	99.8	99.8	99.5	99.7	99.9	99.9	99.7	99.8	99.7	99.6	99.6	99.3	99.8	99.4	99.2	99.6	99.7	99.0	99.2	99.4	99.0	99.5	99.1
r06	88.3	85.3	83.9	85.6	82.2	83.8	87.4	85.3	86.6	84.2	75.4	78.6	77.7	84.1	78.1	75.4	76.7	76.8	68.4	72.6	73.8	70.9	72.7
r07	78.8	78.0	77.2	77.6	76.0	72.8	71.2	70.2	68.0	67.1	68.1	70.1	59.7	58.9	69.2	70.3	65.4	60.3	55.1	44.4	41.5	44.8	28.2
r08	99.0	98.9	99.1	99.0	99.0	98.9	98.8	99.0	99.1	98.9	98.8	99.1	98.9	98.9	99.0	98.9	99.1	98.8	98.7	99.1	98.8	98.5	98.7
r09	98.4	98.4	97.9	98.1	98.2	98.1	97.7	98.0	98.1	98.0	97.9	97.3	97.3	97.8	97.9	97.1	95.0	96.6	97.5	97.4	96.9	97.0	94.9
r10	86.5	86.2	86.6	87.2	87.0	87.4	88.4	87.9	88.3	88.1	87.6	87.2	86.9	86.0	86.8	86.3	86.0	86.7	87.3	87.5	86.6	85.7	85.3

Table 10
F1 and PPV accuracy results of CEC2022EA4eig on ten recordings (r01-r10) with a window size of D20 to D42. The population size is set to 15.

F1	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42
r01	98.7	98.7	98.6	98.8	98.8	98.4	98.8	99.1	98.9	98.9	98.8	98.1	98.9	99.1	98.6	98.7	98.4	98.8	98.5	99.0	98.6	98.4	98.1
r02	98.8	98.5	98.8	98.5	98.4	98.3	98.4	98.5	98.3	98.4	98.3	98.2	98.2	97.9	98.1	98.2	98.1	98.3	98.3	98.2	98.3	98.3	98.4
r03	84.9	78.3	87.2	85.0	83.5	87.0	73.4	89.0	80.2	80.7	74.5	69.6	72.1	74.4	71.7	66.0	71.6	66.0	64.0	62.5	62.2	65.5	68.9
r04	77.4	78.0	71.7	74.3	73.5	67.5	72.0	73.1	69.6	73.6	70.2	72.2	71.1	70.3	72.1	64.6	72.4	64.8	67.5	51.8	50.4	44.9	32.9
r05	99.9	99.8	100.0	99.8	100.0	100.0	99.9	99.9	99.9	99.6	99.6	99.8	99.9	99.8	99.7	99.8	99.7	99.7	99.7	99.5	99.4	99.7	99.7
r06	90.2	90.8	86.6	88.3	91.2	84.9	85.8	83.3	89.7	86.6	87.1	81.5	86.8	77.4	85.1	86.9	87.3	81.1	72.7	84.5	80.2	80.8	79.3
r07	80.3	79.5	78.5	75.8	75.3	74.5	73.9	67.1	69.5	69.5	69.3	67.4	63.4	69.8	67.4	69.2	68.0	67.9	59.5	53.4	47.6	35.7	33.4
r08	99.2	99.1	99.2	99.2	99.3	99.1	99.1	99.2	99.2	99.1	99.2	99.2	99.1	99.2	99.2	99.0	99.1	99.2	99.2	99.2	98.9	99.2	99.1
r09	98.1	98.6	98.7	98.7	98.5	98.2	98.4	98.4	98.1	98.1	98.1	98.2	98.5	98.4	96.9	97.6	98.1	98.2	98.1	98.3	97.8	98.0	97.8
r10	88.5	88.3	89.5	89.6	90.0	89.9	90.9	90.9	90.6	89.9	89.0	89.7	88.5	89.3	90.2	89.6	90.1	89.1	90.0	89.5	89.5	88.6	87.5
PPV	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	D39	D40	D41	D42
r01	98.6	98.6	98.5	98.6	98.7	98.3	98.7	99.0	98.8	98.8	98.8	98.1	98.8	99.0	98.4	98.6	98.3	98.7	98.3	98.9	98.5	98.2	98.0
r02	98.6	98.2	98.6	98.3	98.2	98.0	98.1	98.3	98.1	98.1	98.1	97.9	98.0	97.6	97.8	98.0	97.8	98.0	98.0	97.9	98.1	98.0	98.2
r03	85.1	78.8	87.4	85.3	83.8	87.0	73.6	89.1	80.9	80.6	74.6	69.9	73.8	75.3	71.1	67.6	72.3	67.3	65.6	63.8	63.6	66.8	70.5
r04	77.1	77.5	70.9	73.6	72.6	66.1	70.9	72.1	68.5	72.4	69.0	71.2	69.9	69.4	71.2	63.6	71.4	63.9	66.5	50.8	49.7	44.3	32.5
r05	99.9	99.8	100.0	99.8	99.9	99.9	99.9	99.9	99.9	99.5	99.6	99.8	99.9	99.9	99.6	99.8	99.6	99.7	99.7	99.5	99.4	99.6	99.7
r06	90.0	90.5	86.5	88.0	91.1	85.1	85.9	83.4	89.6	86.8	87.0	82.1	86.7	77.5	85.3	86.9	87.2	81.3	73.8	84.5	80.2	81.0	79.9
r07	79.6	78.8	77.7	74.8	74.1	73.5	72.7	65.8	68.3	68.2	68.0	66.1	61.9	68.5	66.0	68.0	66.8	66.6	58.4	52.6	46.5	34.9	32.7
r08	99.0	99.0	99.1	99.1	99.2	99.0	99.0	99.1	99.1	99.0	99.0	99.0	98.9	98.9	98.9	98.7	98.9	98.9	99.0	99.0	98.7	98.9	98.8
r09	97.9	98.5	98.6	98.6	98.3	98.0	98.3	98.2	97.8	97.9	97.9	98.0	98.3	98.3	96.5	97.4	97.8	98.0	97.9	98.1	97.7	97.8	97.6
r10	86.3	86.0	87.2	87.2	87.7	87.6	88.6	88.6	88.4	87.7	86.7	87.4	86.2	86.9	87.9	87.4	87.8	86.8	87.6	87.2	87.1	86.3	85.2

SOTA MA application

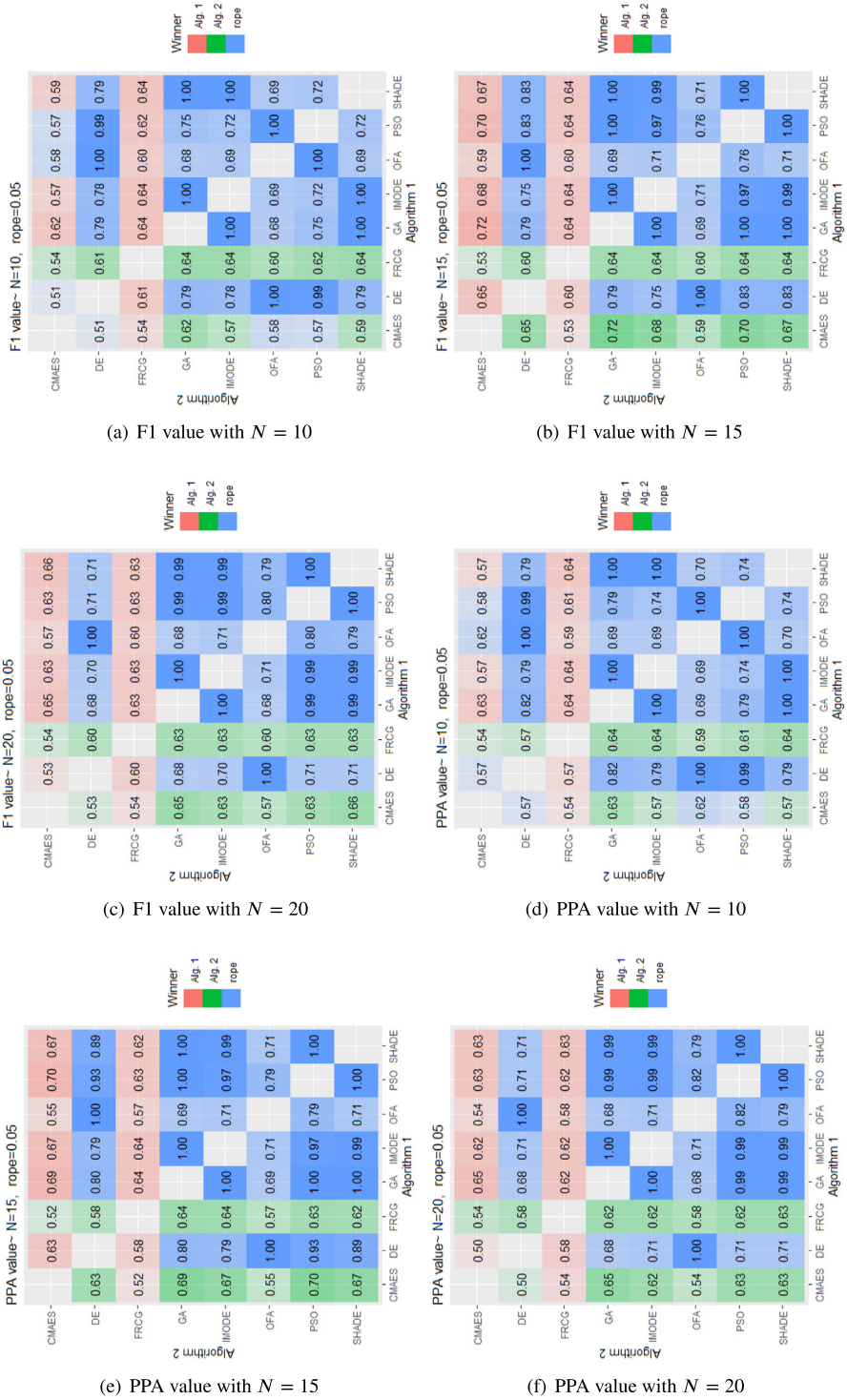
We selected eight classic and diverse MAs as the base optimization algorithm and test them individually through the proposed framework. In fact, any MA can be employed in this framework through the designed interface. We generalize this adaptive filter process as an abstract optimization problem object so that users can employ their desired algorithm without effort. To run one's algorithm with parameters (i.e., population size, iteration number, …) and solve this designed problem, that is it.

We conducted the experiment with the result by the ninth chosen algorithm, CEC2022EA4eig [67], which ranks first in the competition of CEC2022 this year https://wcci2022.org/, and focuses on Single Objective Bound Constrained Optimization. As for the time limits (we add this subsection as the review concerns), we ran the experiment with a population size of 10 and 15, respectively, and the window size ranging from 20 to 42.

Tables 9 and 10 show the F1 and PPV results obtained by the proposed framework embedded with CEC2022EAeig algorithm, with a population size of 10 and 15, respectively. CEC2022EAeig obtained over 98% accuracy on r01, r02, r05, r09 and r08, over 80% accuracy on r03 and r10. While it performed poorly on r04 and r07 recordings with 78% accuracy. There is no significant difference in the overall performance of CEC2022EAeig algorithm compared to other classical algorithms, indicating the robustness of the proposed framework to the selected MA. This phenomenon might imply that the MA is not the primary target for improving comprehensive performance.

5. Conclusion

This work explores population-based optimization algorithms' application and potential problems in the non-invasive fECG extraction field. Hence, this paper proposes a generalization framework comprised of pre-processing, adaptive filters assisted by MA, and post-processing operation. Considering the expansion of general use in real-life situations, we design this framework by eliminating the influence of individual-related parameters; that is to say, we experimented with the framework by applying the (any) algorithm with its default parameter settings in its



(a) F1 value with $N = 10$ (b) F1 value with $N = 15$

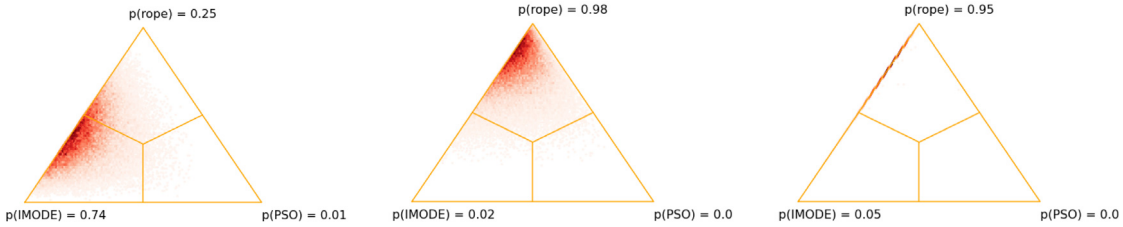
(c) F1 value with $N = 20$ (d) PPA value with $N = 10$

(e) PPA value with $N = 15$ (f) PPA value with $N = 20$

Fig. 7. Bayesian test with *rope*= 0.05 on average results by eight chosen algorithms. a value in a square cell indicates the posterior probability density of winning.

implementation and obtain improved results with the help of post-processing. With the extensive and comprehensive experiment, the results and this work can inform the performance of population-based optimization algorithms MA assisted signal extraction methods and pave the path for future works.

From the experiment above, we can conclude the following points: *(1)* Generally, for the data in this work, IMODE and PSO perform better than other algorithms that can produce higher F1 results on fECG extraction problems. IMODE performs slightly more consistently better (analysis by Fig. 7). *(2)* On recordings r01, r02, r03, r05, r08, and r09, most of the selected algorithms are competitive. They have little influence by the population size and window size changes, where population size varies from 10 to 20, and window size varies from 21 to 42 (analysis by Fig. 5).



(a) F1 value with $N = 10$ (b) F1 value with $N = 15$ (c) F1 value with $N = 20$

Fig. 8. Bayesian test with *rope*= 0.01 on average results of window size varying from 21 to 42 by optimization algorithms IMODE and PSO on certain population size; the value p( ) is the posterior probability density of winning.

Table A.1
The results of LMS on processed data of mECG and aECG* by Fig. 2.

	F1	PPV	SE	MAE	TP	FN	FP
r01	0.727	0.7297	0.7252	4.5824	467	177	173
r02	0.855	0.8591	0.85	10.4759	561	99	92
r03	0.34	0.4455	0.2749	23.9255	188	496	234
r04	0.333	0.3678	0.3033	16.1615	192	441	330
r05	0.757	0.7663	0.7473	7.0436	482	163	147
r06	0.368	0.4629	0.3052	25.4951	206	469	239
r07	0.426	0.4494	0.4048	16.3676	253	372	310
r08	0.847	0.8479	0.8466	4.096	552	100	99
r09	0.72	0.7253	0.7154	7.8702	470	187	178
r10	0.199	0.2224	0.1797	22.3097	113	516	395

Table A.2
Comparative evaluation for fetal extraction and estimation Vs. state-of-the-art methods.

Methods	Years	Recording	Se	PPV	F1
DPSS [68]	2022	10	96.5	97.98	97.24
STFT+GAN [9]	2021	5	90.32	89.79	90.05
CycleGAN [8]	2021	5	99.4	99.6	99.7
RCED-NET [69]	2019	9	96.06	92.25	94.1
STFT+NMF [70]	2019	5	95.3	94.6	94.8
SVD+SW [71]	2017	2	99.36	99.52	99.44
Proposed method	2023	2	98.9	98.6	98.8
		5	100	100	100
		9	99.4	99.2	99.3
		10	94.1	90.1	91.9

(3) For recording r04, r07, r06, and r10, the window size affects the performance of the population-based optimization algorithm (analysis by Fig. 6), indicating the population size is an important factor that affects the performance (analysis by Fig. 6(f)). *(4)* The amplitude tuning is necessary for an optimization-based filter. Otherwise, the filter assisted by MA cannot yield consistent results due to the various hyperparameters.

Meanwhile, we observe that some potential appearances cannot be overlooked.

- The MA may not be a vital target that improves the filter's performance based on LMS. This framework experimented on diverse MAs, and the results cannot be considered significantly different, except for a non-metaheuristic algorithm.
- Those recordings, such as r01, r02, or r05, are less noise-polluted signals which can be solved well by MA than those much noise-polluted signals.

In this work, we experimented with ten signals. We plan to experiment with more data in future work. More importantly, it would be helpful to the model application if we could find the potential problem, such as the algorithm's sensitivity to the window size and population size on specific data and which type of algorithm suits general data. Moreover, the work proposes an optimization-based filter for fECG extraction based on the LMS algorithm principle. The feasibility and effectiveness of other filter methods are still unknown. In contrast, the fitness function designed in our work is simple. Designing a better objective function is challenging, and the function should not be limited to single-objective problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We use open ADFECGDB database source from https://physionet.org.

Appendix

Table A.1 shows the results of fECG extraction by the LMS method. The data is mECG and reference aECG* obtained after the pre-processing process. The window size is set to 30, and the step

size and the equation are as follows:

$$\begin{cases} R & = M(mECG) * M(mECG)^T : \text{ self-correlated matrix of input signal} \\ lambda\_max & = max(eig(R)) : \text{ upper threshold of convergence parameter} \\ mu & = 0.1 * (1/lambda_max) : \text{ parameter } 0 < \text{mu} < 1/\text{lambda\_max} \end{cases} \quad (6)$$

References

[1] R. Jaros, R. Martinek, R. Kahankova, Non-adaptive methods for fetal ECG signal processing: A review and appraisal, Sensors 18 (11) (2018) 3648.
[2] A. Mohammed Kaleem, R.D. Kokate, A survey on FECG extraction using neural network and adaptive filter, Soft Comput. 25 (6) (2021) 4379–4392.
[3] I. Mousavian, M.B. Shamsollahi, E. Fatemizadeh, Noninvasive fetal ECG extraction using doubly constrained block-term decomposition, Math. Biosci. Eng 17 (1) (2019) 144–159.
[4] L. Yuan, Z. Zhou, Y. Yuan, S. Wu, An improved fastica method for fetal ECG extraction, Comput. Math. Methods Med. 2018 (2018).
[5] R. Jaros, R. Martinek, R. Kahankova, J. Koziorek, Novel hybrid extraction systems for fetal heart rate variability monitoring based on non-invasive fetal electrocardiogram, IEEE Access 7 (2019) 131758–131784.
[6] E. Merdjanovska, A. Rashkovska, Comprehensive survey of computational ECG analysis: Databases, methods and applications, Expert Syst. Appl. (2022) 117206.
[7] D.P. Sargam, J. Sahambi, A comparative survey on removal of MECG artifacts from FECG using ICA algorithms, in: International Conference on Intelligent Sensing and Information Processing, 2004. Proceedings of, IEEE, 2004, pp. 88–91.
[8] M.R. Mohebbian, S.S. Vedaei, K.A. Wahid, A. Dinh, H.R. Marateb, K. Tavakolian, Fetal ECG extraction from maternal ECG using attention-based cyclegan, IEEE J. Biomed. Health Inf. 26 (2) (2021) 515–526.
[9] W. Zhong, W. Zhao, Fetal ECG extraction using short time Fourier transform and generative adversarial networks, Physiol. Meas. 42 (10) (2021) 105011.
[10] Y. Zhang, S. Yu, Single-lead noninvasive fetal ECG extraction by means of combining clustering and principal components analysis, Med. Biol. Eng. Comput. 58 (2) (2020) 419–432.
[11] K. Barnova, R. Martinek, R. Jaros, R. Kahankova, A. Matonia, M. Jezewski, R. Czabanski, K. Horoba, J. Jezewski, A novel algorithm based on ensemble empirical mode decomposition for non-invasive fetal ECG extraction, PLoS One 16 (8) (2021) e0256154.
[12] C.N. Praneeth, J.D.K. Abel, D. Samiappan, R. Kumar, S.P. Kumar, P.V. Nitin, A comparison on variants of lms used in fir adaptive noise cancellers for fetal ecg extraction, Biomed. Eng. Appl. Basis Commun. 32 (04) (2020) 2050026.
[13] B. Vasudeva, P. Deora, P.M. Pradhan, S. Dasgupta, Efficient implementation of LMS adaptive filter-based FECG extraction on an FPGA, Healthc. Technol. Lett. 7 (5) (2020) 125–131.
[14] S. Sarafan, T. Le, M.P. Lau, A. Hameed, T. Ghirmai, H. Cao, Fetal electrocardiogram extraction from the mother's abdominal signal using the ensemble Kalman filter, Sensors 22 (7) (2022) 2788.
[15] J.D.K. Abel, D. Samiappan, R. Kumar, S.P. Kumar, Multiple sub-filter adaptive noise canceller for fetal ECG extraction, Procedia Comput. Sci. 165 (2019) 182–188.
[16] B. Galván, D. Greiner, J. Periaux, M. Sefrioui, G. Winter, Parallel evolutionary computation for solving complex CFD optimization problems: a review and some nozzle applications, in: Parallel Computational Fluid Dynamics 2002, Elsevier, 2003, pp. 573–604.
[17] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.
[18] V. Feoktistov, Differential Evolution, Springer, 2006.
[19] M. Clerc, Particle Swarm Optimization, Vol. 93, John Wiley & Sons, 2010.
[20] M.A.S. Ali, X. Zeng, A novel technique for extraction foetal electrocardiogram using adaptive filtering and simple genetic algorithm, Am. J. Biostat. 1 (2) (2010) 75.
[21] K. Nazarpour, S. Ebadi, S. Sanei, Fetal electrocardiogram signal modelling using genetic algorithm, in: 2007 IEEE International Workshop on Medical Measurement and Applications, IEEE, 2007, pp. 1–4.
[22] M. Nasiri, K. Faez, Extracting fetal electrocardiogram signal using ANFIS trained by genetic algorithm, in: 2012 International Conference on Biomedical Engineering, ICoBE, IEEE, 2012, pp. 197–202.
[23] M. Talha, M. Guettouche, A. Bousbia-Salah, Combination of a FIR filter with a genetic algorithm for the extraction of a fetal ECG, in: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, IEEE, 2010, pp. 1756–1759.
[24] A. Alipour, F. Hardalac, Application of genetic algorithms in fuzzy wavelet neural network for fetal electrocardiogram extraction, Int. J. Med. Eng. Inform. 4 (2) (2012) 176–183.
[25] S.A. Billings, H.-L. Wei, A new class of wavelet networks for nonlinear system identification, IEEE Trans. Neural Netw. 16 (4) (2005) 862–874.
[26] D. Panigrahy, M. Rakshit, P. Sahu, An efficient method for fetal ECG extraction from single channel abdominal ECG, in: 2015 International Conference on Industrial Instrumentation and Control, ICIC, IEEE, 2015, pp. 1083–1088.
[27] S. Kockanat, S. Kockanat, Analysis and extraction of fetal electrocardiogram signal with adaptive filtering using differential evolution algorithm, Cumhuriyet Sci. J. 39 (1) (2018) 294–302.
[28] M.S. Jibia, A.U. Jibia, Fetal electrocardiogram extraction using moth flame optimization (MFO)-based adaptive filter, J. Adv. Sci. (2) (2021) 303–312.
[29] M. Akhavan-Amjadi, Fetal electrocardiogram modeling using hybrid evolutionary firefly algorithm and extreme learning machine, Multidimens. Syst. Signal Process. 31 (1) (2020) 117–133.
[30] A. Raj, J. Brablik, R. Kahankova, R. Jaros, K. Barnova, V. Snasel, S. Mirjalili, R. Martinek, Nature inspired method for noninvasive fetal ECG extraction, 2022.
[31] K. Pollard, J. Banerjee, X. Doan, J. Wang, X. Guo, R. Allaway, S. Langmead, B. Slobogean, C.F. Meyer, D.M. Loeb, et al., A clinically and genomically annotated nerve sheath tumor biospecimen repository, Sci. Data 7 (1) (2020) 1–9.
[32] M.S. Jibia, Comparison of moth flame and fruit fly optimization algorithms in fetal electrocardiogram extraction using adaptive filtering approach, 2020.
[33] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.
[34] B. Widrow, M.E. Hoff, Adaptive Switching Circuits, Stanford Univ Ca Stanford Electronics Labs, 1960.
[35] J. Wang, S. Lu, S.-H. Wang, Y.-D. Zhang, A review on extreme learning machine, Multimedia Tools Appl. (2021) 1–50.
[36] X.-S. Yang, A. Slowik, Firefly algorithm, in: Swarm Intelligence Algorithms, CRC Press, 2020, pp. 163–174.
[37] A. Wald, Sequential Analysis, Courier Corporation, 2004.
[38] R. Kahankova, M. Mikolasova, R. Martinek, Optimization of adaptive filter control parameters for non-invasive fetal electrocardiogram extraction, PLoS One 17 (4) (2022) e0266807.
[39] J.V. Stone, Independent Component Analysis: a Tutorial Introduction, MIT Press, 2004.
[40] S. Haykin, B. Widrow, Least-Mean-Square Adaptive Filters, Vol. 31, John Wiley & Sons, 2003.
[41] S.D. Stearns, Of aldapfive signal processing, 1985.
[42] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P.N. Suganthan, C.A.C. Coello, F. Herrera, Bio-inspired computation: Where we stand and what's next, Swarm Evol. Comput. 48 (2019) 220–250.
[43] G.-Y. Zhu, W.-B. Zhang, Optimal foraging algorithm for global optimization, Appl. Soft Comput. 51 (2017) 294–313.
[44] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients, Comput. J. 7 (2) (1964) 149–154.
[45] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[46] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Ieee, 1995, pp. 39–43.
[47] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications To Biology, Control, and Artificial Intelligence, MIT Press, 1992.
[48] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 71–78.
[49] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[50] K.M. Sallam, S.M. Elsayed, R.K. Chakrabortty, M.J. Ryan, Improved multi-operator differential evolution algorithm for solving unconstrained problems, in: 2020 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2020, pp. 1–8.
[51] R. Martinek, R. Kahankova, J. Jezewski, R. Jaros, J. Mohylova, M. Fajkus, J. Nedoma, P. Janku, H. Nazeran, Comparative effectiveness of ICA and PCA in extraction of fetal ECG from abdominal signals: Toward non-invasive fetal monitoring, Front. Physiol. 9 (2018) 648.
[52] R. Kahankova, R. Martinek, R. Jaros, K. Behbehani, A. Matonia, M. Jezewski, J.A. Behar, A review of signal processing techniques for non-invasive fetal electrocardiography, IEEE Rev. Biomed. Eng. 13 (2019) 51–73.
[53] S. Luo, P. Johnston, A review of electrocardiogram filtering, J. Electrocardiol. 43 (6) (2010) 486–496.
[54] K. Barnova, R. Martinek, R. Jaros, R. Kahankova, K. Behbehani, V. Snasel, System for adaptive extraction of non-invasive fetal electrocardiogram, Appl. Soft Comput. 113 (2021) 107940.
[55] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[56] I. Silva, G.B. Moody, An open-source toolbox for analysing and processing physionet databases in matlab and octave, J.Open Res. Softw. 2 (1) (2014).
[57] F. Liu, C. Liu, X. Jiang, Z. Zhang, Y. Zhang, J. Li, S. Wei, Performance analysis of ten common QRS detectors on different ECG application cases, J. Healthc. Eng. 2018 (2018).
[58] E.M. Symonds, A. Chang, D. Sahota, Fetal Electrocardiography, World Scientific, 2001.
[59] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, IEEE Trans. Biomed. Eng. (3) (1985) 230–236.
[60] J. Behar, J. Oster, G.D. Clifford, Combining and benchmarking methods of foetal ECG extraction without maternal or scalp electrode data, Physiol. Meas. 35 (8) (2014) 1569.
[61] J. Jezewski, A. Matonia, T. Kupka, D. Roj, R. Czabanski, Determination of fetal heart rate from abdominal signals: evaluation of beat-to-beat accuracy in relation to the direct fetal electrocardiogram, Biomed. Tech. Biomed. Eng. 57 (5) (2012) 383–394.
[62] M. Kotas, J. Jezewski, K. Horoba, A. Matonia, Application of spatio-temporal filtering to fetal electrocardiogram enhancement, Comput. Methods Programs Biomed. 104 (1) (2011) 1–9.
[63] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[64] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm Evol. Comput. 54 (2020) 100665.
[65] M.J. Zyphur, F.L. Oswald, Bayesian estimation and inference: A user's guide, J. Manag. 41 (2) (2015) 390–420.
[66] S. Herbold, Autorank: A python package for automated ranking of classifiers, J. Open Source Softw. 5 (48) (2020) 2173.
[67] P. Bujok, P. Kolenovsky, Eigen crossover in cooperative model of evolutionary algorithms applied to CEC 2022 single objective numerical optimisation, in: 2022 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2022, pp. 1–8.
[68] A. Shokouhmand, N. Tavassolian, Fetal electrocardiogram extraction using dual-path source separation of single-channel non-invasive abdominal recordings, IEEE Trans. Biomed. Eng. 70 (1) (2022) 283–295.
[69] W. Zhong, L. Liao, X. Guo, G. Wang, Fetal electrocardiography extraction with residual convolutional encoder–decoder networks, Australas. Phys. Eng. Sci. Med. 42 (2019) 1081–1089.
[70] D. Gurve, S. Krishnan, Separation of fetal-ECG from single-channel abdominal ECG using activation scaled non-negative matrix factorization, IEEE J. Biomed. Health Inf. 24 (3) (2019) 669–680.
[71] N. Zhang, J. Zhang, H. Li, O.O. Mumini, O.W. Samuel, K. Ivanov, L. Wang, A novel technique for fetal ECG extraction using single-channel abdominal recording, Sensors 17 (3) (2017) 457.