



Performance Investigation of Meta-Heuristic Algorithms in Estimation of ECG Dynamic Model Parameters

Delavar Matanaq, Javad¹ / Danandeh Hesar, Hamed^{2*} / Ahmadi Fam, Mohammad Hadi¹

¹ - B.Sc. Student, Faculty of Biomedical Engineering, Sahand University of Technology, Tabriz, Iran

² - Assistant Professor, Faculty of Biomedical Engineering, Sahand University of Technology, Tabriz, Iran

ARTICLE INFO

DOI: 10.22041/ijbme.2023.1999546.1838

Received: 5 April 2023

Revised: 3 August 2023

Accepted: 12 August 2023

KEYWORDS

Metaheuristic Optimization
ECG Processing
ECG Dynamic Model

ABSTRACT

In recent years, model-based ECG processing algorithms have been successfully developed in various fields of ECG processing. The calculation of ECG dynamic model (EDM) is a crucial step for these methods. The EDM parameters can be calculated using optimization algorithms. One of the popular optimization methods in this field is an offline nonlinear method in which users have to manually select points on ECG signal in order to calculate EDM parameters. The objective function used in this algorithm is a complex function which is hard to optimize. In this paper an automatic optimization algorithm is proposed which uses meta-heuristic optimization algorithms to calculate EDM parameters. In this algorithm, we don't need to select points manually. In addition, the objective function in this algorithm is broken in to several simple objective functions which makes the optimization more accurate. Meta-heuristic optimization algorithms may perform successfully on some optimization problems while failing on others. As a result, a specific algorithm cannot be considered the best optimizer for all optimization problems. For this reason, in this paper, the performances of nine popular meta-heuristic algorithms such as particle swarm optimization, artificial bee colony, cuckoo search, etc are investigated. In this paper, 200 ECG segments from different records of the MIT-BIH Normal Sinus Rhythm Database (NSRDB) have been selected for evaluation. The duration of each segment was 30 seconds. The EDM parameters for each segment were calculated using the aforementioned optimization algorithms. For evaluation, the similarities between the original signals and the synthetic ECG signals were inspected for each optimization algorithm. These synthetic signals were created using the calculated EDM parameters. The similarity results showed that the water evaporation optimization (WEO), teaching learning-based optimization (TLBO), and cuckoo's search (CS) algorithms achieved better results compared with other methods.

*Corresponding Author

Address: Faculty of Biomedical Engineering, Sahand University of Technology, Tabriz, Iran

Postal Code: 51335-1996

E-Mail: danandeh@sut.ac.ir

Tel: +98-41-33458413





بررسی عمل کرد الگوریتم‌های فراابتکاری در استخراج پارامترهای مدل دینامیکی سیگنال ECG

دلاورمتنق، جواد^۱ / داننده حصار، حامد^{۲*} / احمدی فام، محمدهادی^۱

^۱ - دانشجوی کارشناسی، دانشکده‌ی مهندسی پزشکی، دانشگاه صنعتی سهند، تبریز، ایران
^۲ - استادیار، گروه بیوالکترونیک، دانشکده‌ی مهندسی پزشکی، دانشگاه صنعتی سهند، تبریز، ایران

مشخصات مقاله

شناسه‌ی دیجیتال: 10.22041/ijbme.2023.1999546.1838

پذیرش: ۲۱ مرداد ۱۴۰۲

بازنگری: ۱۲ مرداد ۱۴۰۲

ثبت در سامانه: ۱۶ فروردین ۱۴۰۲

چکیده

واژه‌های کلیدی

در سال‌های اخیر استفاده از الگوریتم‌های مدل-پایه برای پردازش سیگنال ECG رواج گسترده‌ای یافته است. استخراج مدل دینامیکی ECG یکی از مراحل مهم در این الگوریتم‌ها است که تاثیر مستقیمی بر عمل کرد آن‌ها دارد. پارامترهای موجود در این مدل را می‌توان با استفاده از الگوریتم‌های بهینه‌سازی محاسبه کرد. یکی از متداول‌ترین الگوریتم‌ها در این زمینه یک الگوریتم غیرخطی آفلاین است که برای تقریب خوب مدل و پارامترهای آن به نقاطی از سیگنال ECG نیاز دارد که باید توسط کاربر به صورت دستی انتخاب شود. علاوه بر مشکل فوق، تابع هدف در این الگوریتم یک تابع پیچیده است که در صورت انتخاب نادرست نقاط مناسب برای بهینه‌سازی، خروجی مناسبی نخواهد داشت. در این مقاله یک الگوریتم جدید خودکار مبتنی بر الگوریتم‌های فراابتکاری معرفی شده است که نیازی به انتخاب دستی نقاط برای مدل‌سازی ECG ندارد. این الگوریتم پیشنهادی به دلیل ساده‌سازی فرایند بهینه‌سازی از دقت بالایی نسبت به الگوریتم بهینه‌سازی غیرخطی آفلاین مورد اشاره برخوردار است. از آنجا که یک الگوریتم فراابتکاری ممکن است در برخی از مسائل بهینه‌سازی موفق و در برخی دیگر ناموفق عمل کند، در این مقاله عمل کرد ۹ الگوریتم فراابتکاری متداول مانند ازدحام ذرات، کلونی زنبور عسل، جست‌وجوی فاخته و ... در استخراج پارامترهای مدل دینامیکی ECG مورد بررسی قرار گرفته است. جهت ارزیابی الگوریتم‌ها از ۲۰۰ سیگنال ۳۰ ثانیه‌ای مستخرج از پایگاه داده‌ی ریتم سینوس نرمال فیزیوت استفاده شده است. به منظور ارزیابی عمل کرد الگوریتم‌ها، شباهت سیگنال‌های اصلی با سیگنال‌های مصنوعی ECG که توسط الگوریتم‌های بهینه‌سازی ساخته شده مورد بررسی قرار گرفته است. نتایج بررسی‌ها نشان داده است که سه الگوریتم جست‌وجوی فاخته، بهینه‌سازی مبتنی بر یادگیری و آموزش و بهینه‌سازی تخییر آب بهترین عمل کرد را در استخراج پارامترهای مدل دینامیکی ECG دارند. نتایج حاصل از این مطالعه نشان داده که خطای میانگین مربعات (MSE) الگوریتم پیشنهادی با استفاده از سه الگوریتم فراابتکاری فوق به ترتیب ۱/۵۰، ۱/۴۳ و ۱/۴۰ بوده در حالی که این مقدار برای الگوریتم بهینه‌سازی غیرخطی آفلاین برابر با ۴/۸۲ است.

*نویسنده‌ی مسئول

نشانی: گروه بیوالکترونیک، دانشکده‌ی مهندسی پزشکی، دانشگاه صنعتی سهند، تبریز، ایران

تلفن: ۰۹۸-۴۱-۳۳۴۵۸۴۱۳

پست الکترونیک: danandeh@sut.ac.ir

کد پستی: ۵۱۳۳۵-۱۹۹۶



۱- مقدمه

سیگنال ECG شامل اطلاعات بسیار مهم از وضعیت قلب و یکی از مهم‌ترین معیارهای پزشکان در تشخیص بیماری‌های قلبی و هم‌چنین سلامت یک شخص است. با گسترش نگران‌کننده‌ی بیماری‌های قلبی به دلیل تغییر شیوه‌ی زندگی و مرگ‌ومیرهای ناشی از آن‌ها، تفسیر این سیگنال و استفاده از اطلاعات مفید آن در تشخیص و درمان بیماری‌ها اهمیتی روزافزون یافته است. در سال‌های اخیر روش‌های مدل-پایه و مبتنی بر فیلترهای بیزی خطی قابلیت مناسبی را از خود نسبت به سایر روش‌ها در زمینه‌ی حذف نویز، قطعه‌بندی و تشخیص محل ناهنجاری در سیگنال ECG نشان داده‌اند [۱-۱۲]. در مقاله‌ی ثامن‌ی و هم‌کارانش با استفاده از فیلتر کالمن گسترش یافته^۱ (EKF) یک چارچوب قدرتمند برای حذف نویز ECG پیشنهاد شده است [۱۰]. در پژوهش صیادی و هم‌کارانش نیز با بهبود مدل پیشنهادی، یک چارچوب اصلاح شده مبتنی بر EKF با هدف حذف نویز و قطعه‌بندی معرفی شده است [۱۱]. هم‌چنین در یک تحقیق دیگر قابلیت این چارچوب اصلاح شده برای تشخیص آریتمی‌ها مورد ارزیابی قرار گرفته و با روش‌های دیگر مقایسه شده است [۱۲]. در مقاله‌ی داننده و هم‌کارانش با استفاده از یک فیلتر ذره‌ای حاشیه‌ای کالمن گسترش یافته^۲ (MP-EKF) چارچوب‌های مختلفی برای حذف نویز [۵، ۶] و قطعه‌بندی [۷] سیگنال ECG معرفی شده است.

یکی از مراحل مهم در این روش‌ها استخراج مدل دینامیکی ECG^۳ (EDM) است که تاثیر مستقیمی بر عملکرد آن‌ها در پردازش سیگنال ECG دارد. برای استخراج پارامترهای EDM در اکثر روش‌های مذکور از یک الگوریتم بهینه‌سازی غیرخطی آفلاین [۱۳] استفاده شده است اما این الگوریتم برای عملکرد مناسب به انتخاب دستی نقاط روی سیگنال ECG توسط کاربر نیاز دارد. هم‌چنین محل قرارگیری این نقاط نیز باید طوری تنظیم شود که به نقاط اکسترمم نسبی موج‌های P، Q، R، S و T در سیگنال ECG نزدیک باشد. در صورت عدم انتخاب مناسب نقاط، به دلیل پیچیده بودن تابع هدف برای بهینه‌سازی، ممکن است الگوریتم بهینه‌سازی غیرخطی مذکور عملکرد مناسبی را در استخراج مدل دینامیکی نداشته باشد.

در این مقاله یک روش خودکار برای محاسبه‌ی پارامترهای EDM معرفی شده که مشکلات روش پیشین را برطرف کرده و با استفاده از الگوریتم‌های بهینه‌سازی فراابتکاری^۴ و با ساده‌سازی توابع هزینه‌ی لازم برای بهینه‌سازی، پارامترهای EDM را محاسبه می‌کند.

الگوریتم‌های فراابتکاری به دلیل ماهیت ذاتی عملکردی خود در بهینه‌سازی و تصادفی بودن فرایند انتخاب اولیه‌ی نقاط راه حل خود ممکن است در برخی از مسائل بهینه‌سازی موفق و در برخی دیگر ناموفق عمل کنند [۱۴]. به عبارت دیگر برتری یک الگوریتم فراابتکاری در یک مساله‌ی خاص به معنای برتری و رجحان آن الگوریتم در حالت کلی نیست. به همین دلیل در این مقاله عملکرد ۹ الگوریتم فراابتکاری پرکاربرد در استخراج پارامترهای مدل دینامیکی ECG مورد بررسی قرار گرفته است تا به دو سوال زیر پاسخ داده شود.

۱- آیا در حالت کلی، ساختار پیشنهادی در مقایسه با الگوریتم بهینه‌سازی غیرخطی آفلاین بهتر عمل می‌کند؟

۲- کدام الگوریتم فراابتکاری برای مساله‌ی بهینه‌سازی مورد نظر مناسب است؟

الگوریتم‌های فراابتکاری مورد استفاده در این مقاله شامل الگوریتم کلونی زنبورهای مصنوعی^۵ (ABC)، الگوریتم بیگ‌بنگ-بیگ‌کرانچ^۶ (BB-BC)، بهینه‌سازی مبتنی بر یادگیری و آموزش^۷ (TLBO)، جست‌وجوی فاخته^۸ (CS)، بهینه‌سازی تبادل حرارتی^۹ (TEO)، الگوریتم سیستم ذرات ارتعاشی^{۱۰} (VPS)، الگوریتم رقابتی امپریالیستی^{۱۱} (ICA)، الگوریتم بهینه‌سازی تخییر آب^{۱۲} (WEO) و الگوریتم بهینه‌سازی ازدحام ذرات^{۱۳} (PSO) است.

پایگاه داده‌ی آزمایشی در این مقاله شامل ۲۰۰ سیگنال ECG ۳۰ ثانیه‌ای بوده که از بخش‌ها و ثبت‌های مختلف موجود در پایگاه داده‌ی ریتم سینوس طبیعی MIT-BIH^{۱۴} (NSRDB) موجود در فیزیونت^{۱۵} استخراج شده است [۱۵]. جهت مقایسه‌ی عملکرد الگوریتم‌های بهینه‌سازی، میزان شباهت سیگنال‌های ECG مصنوعی نسبت به سیگنال‌های واقعی برای هر الگوریتم بررسی شده است. این سیگنال‌های ECG مصنوعی توسط EDM-هایی که توسط الگوریتم‌های فراابتکاری محاسبه

^۱ Thermal Exchange Optimization

^۲ Vibrating Particle System

^۳ Imperialist Competitive Algorithm

^۴ Water Evaporation Optimization

^۵ Particle Swarm Optimization

^۶ MIT-BIH Normal Sinus Rhythm Database

^۷ Physionet

^۱ Extended Kalman Filter

^۲ Marginalized Particle Extended Kalman Filter

^۳ ECG Dynamic Model

^۴ Metaheuristic Optimization Algorithms

^۵ Artificial Bee Colony

^۶ Big Bang-Big Crunch

^۷ Teaching Learning Based Optimization

^۸ Cuckoo's Search

و بهره‌برداری بیان‌گر توانایی الگوریتم برای انجام جست‌وجوی محلی در فضای جست‌وجو است. کلید موفقیت یک الگوریتم فراابتکاری در فرایند بهینه‌سازی، حفظ تعادل مناسب بین اکتشاف و بهره‌برداری است [۱۸].

در این بخش ابتدا به صورت اجمالی ۹ الگوریتم فراابتکاری مورد استفاده در این مقاله توضیح داده شده و سپس به معرفی روش پیشنهادی برای استخراج پارامترهای EDM پرداخته شده است.

۲-۱- الگوریتم‌های بهینه‌سازی فراابتکاری

در این بخش به معرفی الگوریتم‌های فراابتکاری مورد استفاده در این مقاله پرداخته شده است. در این الگوریتم‌ها متغیرهای مجهول تابع هدف^۴ (Fit) بدون نیاز به مشتق‌گیری توسط جمعیتی از بردارهای راه حل یافته می‌شود. معمولاً هدف الگوریتم‌های فراابتکاری یافتن مقادیری برای این متغیرها است به طوری که تابع Fit را کمینه سازد. هر کدام از بردارهای راه حل حاوی مقادیر کاندیدا برای متغیرهای مساله هستند. با جایگذاری هر بردار راه حل در تابع هدف، مقداری با نام تابع هدف جریمه شده^۵ (PFit) به دست می‌آید که هر چه کم‌تر باشد امتیاز آن بردار راه حل نسبت به سایر رقبا بیش‌تر است. هر کدام از الگوریتم‌های فراابتکاری معادلات منحصر به فردی برای تولید و اصلاح بردارهای راه حل دارند که در ادامه به معرفی آن‌ها پرداخته شده است.

۲-۱-۱- کلونی زنبورهای مصنوعی

الگوریتم کلونی زنبورهای مصنوعی (ABC) توسط کارابوگا در سال ۲۰۰۵ بر اساس رفتار زنبورهای عسل برای جست‌وجوی غذا توسعه داده شده است [۲۰]. در الگوریتم ABC فرض می‌شود هر راه حل احتمالی یک منبع غذایی بوده و کیفیت شهد آن نشان دهنده‌ی عمل‌کرد آن راه حل است. این منابع غذایی توسط زنبورهای عسل در یک فرایند تکراری با هدف دستیابی به منابع غذایی با شهد بهتر اصلاح می‌شود. الگوریتم ABC با تولید تصادفی تعداد nHB زنبور عسل (بردار راه حل) در فضای جست‌وجو شروع به کار می‌کند. پس از ارزیابی، بر اساس بردار تابع هدف مربوطه (Fit)، تابع هدف جریمه شده (PFit) برای هر زنبور محاسبه می‌شود. در ABC زنبورهای عسل به سه نوع زنبورهای کارگر^۶، تماشاگر^۷ و پیشاهنگ^۸ تقسیم می‌شوند. زنبورها با توجه به وظیفه‌ی خود،

گردیده ساخته شده است. در این مقاله از دو معیار ارزیابی شباهت با نام‌های میانگین مربعات خطا^۱ (MSE) و همبستگی متقابل نرمالیزه (NCC)^۲ استفاده شده است. نتایج آزمایش‌ها نشان داده است که سه الگوریتم CS، TLBO و WEO بهترین عمل‌کرد را در استخراج دینامیک سیگنال‌های ECG دارند. نوآوری‌های این مقاله به شرح زیر است.

۱- برای نخستین بار از الگوریتم‌های فراابتکاری برای استخراج پارامترهای EDM استفاده شده است.

۲- تابع بهینه‌سازی روش غیرخطی آفلاین که یک تابع پیچیده برای بهینه‌سازی بوده شکسته شده و به توابع ساده برای بهینه‌سازی تبدیل شده است.

۳- روش پیشنهادی کاملاً خودکار بوده و نیازی به انتخاب دستی نقاط توسط کاربر ندارد.

در ادامه در بخش ۲ به معرفی الگوریتم‌های فراابتکاری مورد استفاده و همچنین روش پیشنهادی برای استخراج پارامترهای EDM پرداخته شده است. در بخش ۳ عمل‌کرد روش‌های فراابتکاری بررسی شده و بخش آخر به بحث و نتیجه‌گیری در مورد روش پیشنهادی اختصاص داده شده است.

۲- مواد و روش‌ها

الگوریتم‌های بهینه‌سازی فراابتکاری به دلیل مزایایی مانند مفاهیم ساده، پیاده‌سازی آسان، استقلال نسبت به نوع مساله، عدم نیاز به مشتق‌گیری از تابع هدف و کارایی در محیط‌های غیرخطی و فضای جست‌وجوی غیرخطی بسیار محبوب و پرکاربرد هستند [۱۶]. فرایند بهینه‌سازی در الگوریتم‌های فراابتکاری با تولید تعدادی راه حل منتخب تصادفی در محدوده‌ی مجاز برای فضای جست‌وجو آغاز می‌شود. سپس در یک فرایند تکراری، راه‌حل‌های منتخب توسط مراحل الگوریتم بهبود می‌یابد. پس از تکمیل تکرارهای پیاده‌سازی الگوریتم، بهترین راه حل منتخب به عنوان راه حل مساله معرفی می‌شود. اما باید به این نکته اشاره کرد که ماهیت جست‌وجوی تصادفی در الگوریتم‌های فراابتکاری منجر به ارائه‌ی راه حل شبه‌بهینه و نه بهینه‌ی سرتاسری^۳ خواهد شد [۱۷].

در طراحی الگوریتم‌های بهینه‌سازی، دو شاخص اکتشاف و بهره‌برداری نقش مهمی در عمل‌کرد الگوریتم‌های بهینه‌سازی در دستیابی به راه‌حل‌های شبه‌بهینه‌ی مناسب دارند. اکتشاف، توانایی الگوریتم برای انجام جست‌وجوی سرتاسری را نشان داده

^۵ Penalized Objective Function

^۶ Employed Bees

^۷ Onlooker Bees

^۸ Scout Bees

^۱ Mean Squared Error

^۲ Normalized Cross Correlation

^۳ Global Optimum

^۴ Objective Function

آیا منبع غذایی انتخاب شده توسط زنبور ناظر تغییر خواهد کرد یا خیر را کنترل می‌کند. معمولاً نرخ اصلاح الگوریتم برابر با $0/8$ در نظر گرفته می‌شود. در مرحله‌ی زنبور پیشاهنگ، زنبورهای شاغلی که نمی‌توانند منابع غذایی خود را پس از تعداد مشخصی آزمایش (A) تغییر دهند، پیشاهنگ می‌شوند. پارامتر NFE تعداد ارزیابی عمل کرد و MaxNFEs حداکثر تعداد ارزیابی تابع هدف است که به عنوان معیار توقف الگوریتم در نظر گرفته می‌شود. فلوجارت الگوریتم ABC در شکل (۱) ارائه شده است.

۲-۱-۲- بیگ‌بنگ-بیگ‌کرانچ

الگوریتم بیگ‌بنگ-بیگ‌کرانچ برگرفته از نظریه‌ی تکاملی رایج برای منشا جهان یعنی نظریه‌ی بیگ‌بنگ است [۱۹]. طبق این نظریه، در فاز بیگ‌بنگ، ذرات با از دست دادن انرژی به سمت بی‌نظمی کشیده شده در حالی که در فاز بیگ‌کرانچ به سمت یک جهت خاص هم‌گرا می‌شوند. تئوری بیگ‌بنگ شامل دو مرحله است. در مرحله‌ی بیگ‌بنگ یک بی‌نظمی برای ذرات از نظر اتلاف انرژی ایجاد شده و در مرحله‌ی بیگ‌کرانچ ذرات پراکنده به سمت یک جهت خاص سوق داده می‌شوند. روابط مورد استفاده در الگوریتم فراابتکاری BB-BC به شرح زیر است.

$$CM(i) = \sum_{j=1}^{np} \frac{P(j,i)}{PFit(j)} \sum_{j=1}^{np} \left(\frac{1}{PFit(j)} \right) \quad i = 1 \dots np \quad (3)$$

در این رابطه تعداد معینی از ذرات (np) به عنوان جمعیت و ماتریس راه حل‌های کاندید P در نظر گرفته شده است. عمل‌گر هم‌گرایی بر اساس فاز بیگ‌کرانچ را می‌توان به عنوان میانگین وزنی موقعیت‌های راه حل منتخب شناخته شده به عنوان مرکز جرم CM^2 یا موقعیت بهترین راه حل منتخب (bestP) تعریف کرد. بردار PFit(j) تابع هدف جریمه شده برای ذره‌ی j-ام و مقدار P(j,i) پیشنهادی ذره‌ی j-ام برای متغیر i-ام در تابع هدف است. با در نظر گرفتن پارامتری برای محدود کردن اندازه‌ی فضای جست‌وجو (α) و تعریف پارامتر وزن (β) برای ارزش‌دهی متفاوت به CM و bestP در تعیین موقعیت ذره‌ی جدید، رابطه‌ی فاز بیگ‌بنگ به صورت زیر تعریف می‌شود.

$$newP(i) = (\beta \times CM + (1 - \beta) \times bestP) + \frac{rand \times \alpha \times (Ub - Lb)}{NIT} \quad i = 1 \dots np \quad (4)$$

جست‌وجوی الگوریتم ممکن است برخی از ابعاد راه حل منتخب را به خارج از فضای جست‌وجو بکشاند. یک روش که با استفاده از آن متغیرهای خارج از محدوده بازگردانده می‌شود تعریف گام

برای رسیدن به بهترین منبع غذایی، استراتژی‌های مختلفی را انجام می‌دهند که این استراتژی‌ها بر اساس معادلات ریاضی تعریف می‌شوند. به عنوان مثال محل زنبورهای عسل کارگر جدید (newHB) بر اساس فرمول زیر تعیین می‌شود.

$$stepsize = rand_{(i)(j)} \cdot (HB - HB[permute(i)(j)]) \quad (1)$$

$$newHB = HB + stepsize$$

در این رابطه HB مکان زنبورهای عسل کارگر کنونی، stepsize بردار جابه‌جایی مکانی برای تولید مکان‌های جدید برای زنبورهای عسل، permute نماد جایگشت ردیف‌های مختلف و $rand_{(i)(j)}$ یک عدد تصادفی انتخاب شده از توزیع یک‌نواخت پیوسته در بازه $[-1, 1]$ بوده که i تعداد زنبورهای عسل و j تعداد ابعاد مساله (تعداد متغیرهای مساله) است. رابطه‌ی زیر نیز برای تعیین محل جدید زنبورهای تماشاگر تعریف می‌شود.

$$stepsize = rand_{(i)(j)} \cdot (HB_{rws} - HB_i[permute(i)(j)]) \quad (2)$$

$$newHB = \begin{cases} HB_{rws} + stepsize & \text{if } Rand < mr \\ HB_{rws} & \text{otherwise} \end{cases}$$



شکل (۱) - فلوجارت الگوریتم ABC

در این رابطه HB محل کنونی زنبورهای تماشاگر و HB_{rws} احتمال انتخاب یک منبع غذایی بر اساس چرخ رولت^۱ توسط زنبور تماشاگر i-ام است. نرخ اصلاح (mr) برای بهینه‌سازی به عنوان یک پارامتر کنترلی تعریف شده است و این موضوع که

^۱ Center of Mass

^۱ Roulette Wheel

خود دارند که این رویه در مرحله‌ی یادگیرنده دنبال می‌شود. مراحل الگوریتم TLBO به شرح زیر است.

۲-۱-۳-۱- توليد يا آموزش دانش‌آموزان (newL) بر اساس مرحله‌ی معلم

عمل کرد کلاس به عنوان توزیع نرمال نمرات به دست آمده توسط دانش‌آموزان را می‌توان با مقدار میانگین توزیع مشخص کرد. در این مرحله، TLBO با تغییر موقعیت متوسط افراد کلاس به سمت بهترین یادگیرنده که به عنوان معلم در نظر گرفته می‌شود قصد دارد عمل کرد کلاس را بهبود بخشد. در این راستا TLBO فراگیران را با اندازه‌ی گام به معلم به‌روزرسانی می‌کند. این گام بر اساس تفاوت بین موقعیت معلم و میانگین موقعیت تمام دانش‌آموزان و با ضرب در یک بردار تصادفی به دست می‌آید. با در نظر گرفتن میانگین موقعیت دانش‌آموزان در فضای جست‌وجو به عنوان MeanL این مرحله را می‌توان به صورت زیر بیان کرد.

$$\begin{aligned} \text{stepsize}_i &= T - TF_i \times \text{MeanL} \\ \text{newL} &= L + \text{rand}_{i,j} \times \text{stepsize} \\ i &= 1, 2 \dots nL \quad j = 1, 2 \dots nV \end{aligned} \quad (5)$$

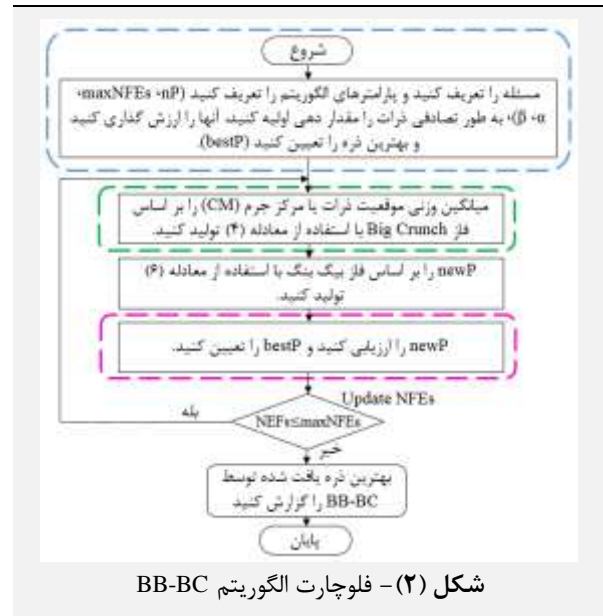
در این رابطه T مکان معلم، nV تعداد متغیرهای تابع هدف و TF_i ضریب یادگیری (پارامتر کنترلی) است که مشخص می‌کند معلم چقدر می‌تواند میانگین دانش کلاس را تغییر دهد. این مقدار می‌تواند ۱ یا ۲ باشد. هم‌چنین nL تعداد دانش‌آموزان و L جامعه‌ی الگوریتم به صورت یک ماتریس ($nL \times nV$) است.

۲-۱-۳-۲- ایجاد یادگیرندگان جدید (newL) یا به‌روزرسانی دانش دانش‌آموزان از طریق تعامل با یک‌دیگر در مرحله‌ی یادگیرنده

در این مرحله هر دانش‌آموز (L_i) با یک دانش‌آموز دیگر (L_{rp}) که به طور تصادفی انتخاب شده است به جز خود او برای بهبود احتمالی دانش تعامل دارد. پس از مقایسه‌ی این دو، دانش‌آموزی در صورت باهوش‌تر بودن ($PFit_i < PFit_{rp}$) به سمت دانش‌آموزی که به طور تصادفی انتخاب شده است منتقل شده و در غیر این صورت از او دور می‌شود. مرحله‌ی یادگیرنده را می‌توان به صورت ریاضی با رابطه‌ی زیر بیان کرد.

$$\begin{aligned} \text{stepsize}_i &= \begin{cases} L_i - L_{rp}, & PFit_i < PFit_{rp} \\ L_{rp} - L_i, & PFit_i \geq PFit_{rp} \end{cases} \\ \text{newL} &= L + \text{rand}_{i,j} \times \text{stepsize} \\ i &= 1, 2 \dots nL \quad j = 1, 2 \dots nV \end{aligned} \quad (6)$$

مجاز با استفاده از کران‌های پایین (Lb) و کران‌های بالا (Ub) است. اندازه‌ی گام بر تعداد تکرارهای الگوریتم یا تعداد فازهای بیگ‌بنگ (NIT) تقسیم شده تا محدوده‌ی جست‌وجوی موثر در مورد بهینه‌ی سرتاسری^۱ یا مرکز جرم ایجاد شود. لازم به ذکر است که الگوریتم BB-BC نیازی به استراتژی جایگزینی ندارد. به عبارت دیگر ذرات موقعیت خود را ترک می‌کنند و مهم نیست که موقعیت فعلی آن‌ها بهتر باشد. در شکل (۲) فلوجارت الگوریتم BB-BC قابل مشاهده است.



شکل (۲) - فلوجارت الگوریتم BB-BC

۲-۱-۳-۳- بهینه‌سازی مبتنی بر یادگیری و آموزش

الگوریتم بهینه‌سازی مبتنی بر یادگیری و آموزش (TLBO) برای اولین بار در سال ۲۰۱۱ توسط راتو و هم‌کارانش [۲۰] بر اساس فرایند یادگیری کلاسیک مدرسه توسعه یافته است. الگوریتم TLBO با جمعیتی از راه‌حل‌های تصادفی به نام دانش‌آموزان یا یادگیرندگان، مقداردهی اولیه می‌شود. باهوش‌ترین دانش‌آموز با بهترین تابع هدف به عنوان معلم در هر تکرار الگوریتم تعیین می‌شود. دانش‌آموزان به طور مکرر برای جست‌وجوی بهینه در دو مرحله ابتدا بر اساس انتقال دانش از معلم (مرحله‌ی معلم) و سپس از تعامل با دانش‌آموزان دیگر (مرحله‌ی یادگیرنده) به‌روز می‌شوند. خروجی الگوریتم همان نمرات دانش‌آموزان و سطح دانش آن‌ها بوده که کیفیت و توانایی معلم در این زمینه بسیار نتیجه‌بخش است. بنابراین معلم هر کلاس بهترین دانش‌آموز آن کلاس را انتخاب می‌کند تا بتواند با راهنمایی سایر دانش‌آموزان نمرات آن‌ها را ارتقا دهد. این فرایند در مرحله‌ی معلم پیگیری می‌شود. هم‌چنین دانش‌آموزان نیز با یادگیری از یک‌دیگر سعی در ارتقای نمرات

^۱ Global Optimum

در این رابطه α پارامتر اندازه‌ی گام است که بیش‌تر از صفر در نظر گرفته شده و بر اساس مقیاس‌های مساله تنظیم می‌شود. هم‌چنین S یک گام تصادفی بر اساس توزیع لوی^۲ است. در الگوریتم جست‌وجوی فاخته، bestNest بدون تغییر نگه داشته شده و راه‌های دیگر برای آن به‌روزرسانی می‌شود. کشف تخم‌های بیگانه برای هر جز از هر راه حل بر حسب ماتریس احتمال کشف (P) به صورت زیر انجام می‌شود.

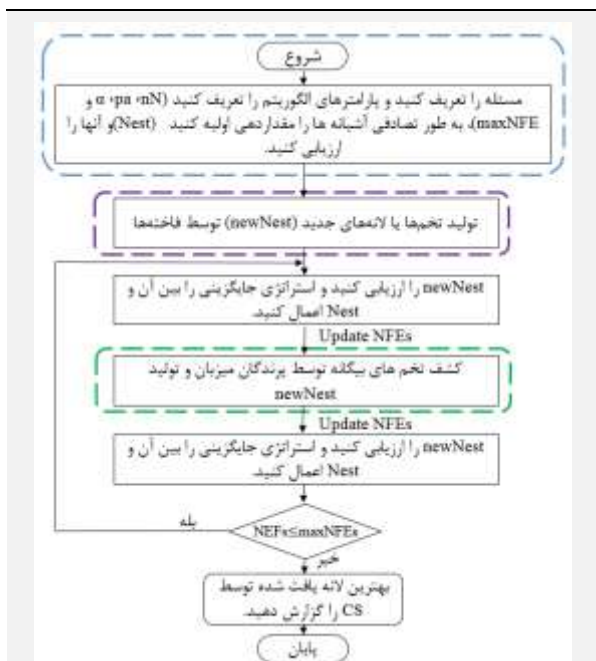
$$P_{(i)(j)} = \begin{cases} 1. & \text{if } rand < pa \\ 0. & \text{if } rand \geq pa \end{cases} \quad (8)$$

در این رابطه pa احتمال کشف و ماتریس P به اندازه‌ی ماتریس Nest است. تخم‌های موجود با توجه به کیفیت آن‌ها با تخم‌های تازه تولید شده از موقعیت فعلی خود از طریق قدم‌های تصادفی^۳ با اندازه‌ی گام مبتنی بر جایگشت تصادفی جایگزین می‌شوند.

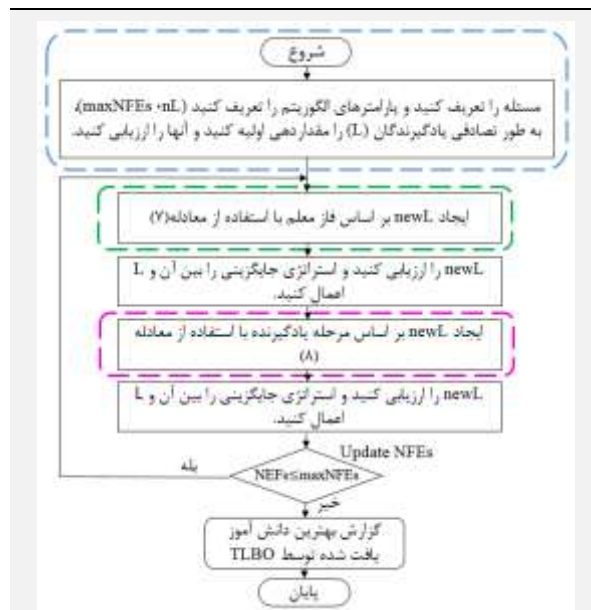
$$stepsize = rand_{(i)(j)} \times (Nest[permute1(i)(j)] - Nest[permute2(i)(j)]) \quad (9)$$

$$newNest = Nest + stepsize \times P$$

در این رابطه permute1 و permute2 توابع جایگشت تصادفی بوده که از آن‌ها برای جایگشت ردیف‌های مختلف اعمال شده در ماتریس Nest استفاده می‌شود. این مرحله توانایی نوع بخشیدن به الگوریتم را تضمین می‌کند.



شکل (۴) - فلوجارت الگوریتم CS

^۲ Random Walk


شکل (۳) - فلوجارت الگوریتم TLBO

۲-۱-۴ - جست‌وجوی فاخته

الگوریتم جست‌وجوی فاخته (CS) توسط یانگ و دب [۲۱] با الهام از رفتار برخی از گونه‌های فاخته و شیوه‌ی پرواز لوی^۱ [۲۲] توسعه یافته است. فاخته‌ها با توانایی‌های شگفت‌انگیزی از جمله انتخاب لانه‌های مناسب تازه تخم‌ریزی شده و حذف تخم‌های موجود که احتمال جوجه شدن تخم‌هایشان را افزایش می‌دهد تخم‌های خود را در لانه‌ی سایر پرندگان می‌گذارند. میزبان از تخم‌ها مراقبت می‌کند با این فرض که تخم‌ها برای خودش است. با این حال اگر تخم فاخته بسیار شبیه به تخم میزبان باشد احتمال کم‌تری برای کشف تخم فاخته وجود دارد. میزبان بخش کوچکی از تخم‌های بیگانه را کشف کرده و با افزودن یک اندازه‌ی گام تصادفی مبتنی بر جایگشت آن‌ها را به‌روز می‌کند. الگوریتم CS با مجموعه‌ای از راه‌های تولید شده به طور تصادفی شروع شده که لانه یا تخم پرندگان میزبان هستند. تعداد لانه‌ها یا تخم‌ها (nN) اولین پارامتر الگوریتم است. پس از ارزیابی، بر اساس بردار تابع هدف مربوطه (Fit)، تابع هدف جریمه شده ($PFit$) برای هر لانه محاسبه می‌شود. محل لانه‌های کنونی یا تخم‌ها به جز بهترین لانه ($bestNest$) بر اساس کیفیت آن‌ها با تخم‌های فاخته‌ی جدید ($newNest$) که با ماتریس راه‌های فعلی ($Nest$) به سمت بهترین لانه در ترکیب با پرواز لوی تولید می‌شود جایگزین می‌گردد.

$$stepsize = rand_{(i)(j)} \times \alpha \times S \times (Nest - bestNest) \quad (7)$$

$$newNest = Nest + stepsize$$

^۱ Lévy Flights

^۲ Levy Distribution

۲-۱-۵- بهینه‌سازی تبادل حرارتی

در مقاله‌ی کاوه و دادرس [۲۳] بر اساس قانون خنک‌سازی نیوتن یک الگوریتم فراابتکاری به نام بهینه‌سازی تبادل حرارتی (TEO) معرفی شده است. طبق قانون سرمایه‌ش نیوتن سرعت تغییر دمای یک جسم متناسب با اختلاف دمای بین جسم و محیط اطراف آن است. الگوریتم TEO هر یک از ذرات را به عنوان یک جسم خنک کننده یا گرم کننده در نظر می‌گیرد و با مرتبط کردن عامل دیگری به عنوان محیط، انتقال و تبادل حرارتی بین آن‌ها اتفاق می‌افتد. دمای جدید جسم به عنوان موقعیت بعدی آن در فضای جست‌وجو در نظر گرفته می‌شود. الگوریتم TEO با مجموعه‌ای از راه‌حل‌های منتخب شروع شده که به طور تصادفی در فضای جست‌وجو ایجاد می‌شود. تعداد اجسام حرارتی^۱ (nTO) در نظر گرفته شده که ماتریس اجسام حرارتی (TO) را تشکیل می‌دهند. پس از ارزیابی، بر اساس بردار تابع هدف مربوطه (Fit)، تابع هدف جریمه شده (PFit) تولید می‌شود. برای فرایند انتقال حرارت باید اجسام بر اساس تابع هدف جریمه شده به طور صعودی مرتب شده و به دو گروه با تعداد اجسام یکسان دسته‌بندی شوند. بدین ترتیب یک راه حل بهتر به عنوان یک جسم گرم‌تر شناخته می‌شود. هر جسم گرم یا سرد به سمت دمای محیط پیش می‌رود. از این رو با در نظر گرفتن نیمه‌ی اول به عنوان اجسام خنک کننده، نیمه‌ی دوم به عنوان اجسام محیطی و برعکس با در نظر گرفتن نیمه‌ی دوم به عنوان اشیای گرمایشی، نیمه‌ی اول به عنوان اشیای محیطی^۲ در نظر گرفته می‌شود. هر جسم (گرم یا سرد) به سمت جسم محیطی مربوطه (envTO) به شرح زیر حرکت می‌کند.

$$newTO(i) = envTO(i) + (TO(i) - envTO(i)) \exp(-\beta(i)t) \quad (10)$$

در این رابطه $newTO(i)$ موقعیت جدید جسم سرد یا گرم شده و t پارامتر زمانی مرتبط با عدد تکرار الگوریتم به شکل زیر است.

$$t = \frac{NITs}{maxNITs} \quad (11)$$

در این رابطه NITs تعداد فعلی تکرار الگوریتم و $maxNITs$ حداکثر تعداد تکرار الگوریتم است که به عنوان معیار توقف الگوریتم در نظر گرفته می‌شود. پارامتر $\beta(i)$ برای هر جسم حرارتی به صورت زیر در نظر گرفته می‌شود.

$$\beta(i) = \frac{PFit(i)}{maxPFit} \quad (12)$$

در این رابطه $maxPFit$ ماکسیمم تابع هدف جریمه شده یا سردترین شی در جمعیت فعلی است. پیش از استفاده از اجسام

حرارتی به عنوان اجسام محیطی در رابطه‌ی (۱۰)، تمام اجسام حرارتی باید طبق مکانیسم زیر به طور تصادفی اصلاح شوند.

$$envTO = (1 - c \times rand)TO \quad (13)$$

$$c = c_1 + c_2(1 - t)$$

در این رابطه c_1 و c_2 پارامترهای کنترلی باینری با مقادیر صفر یا ۱ هستند. ترم $(1-t)$ تصادفی بودن را به صورت خطی کاهش داده و با نزدیک شدن به آخرین تکرارها، بهره‌برداری را افزایش می‌دهد. الگوریتم TEO با مکانیسم دیگری نیز موقعیت اجسام کنونی را تغییر می‌دهد. در این مکانیسم یک پارامتر کنترلی p برای هر جسم در نظر گرفته می‌شود. برای هر جسم این پارامتر با یک عدد تصادفی $rand$ که از توزیع یک‌نواخت بین صفر تا ۱ انتخاب شده مقایسه می‌شود. اگر $rand < p$ باشد آن جسم از فرایند بهینه‌سازی حذف شده و با یک جسم دیگر که به طور تصادفی از فضای جست‌وجو انتخاب شده جایگزین می‌شود. الگوریتم TEO دارای حافظه‌ای است که در آن برخی از بهترین اشیا (TO_M) که تا به حال یافته شده به همراه تابع هدف (Fit_M) و تابع هدف جریمه شده‌ی آن‌ها ($PFit_M$) ذخیره می‌شود. اندازه‌ی این حافظه STO_M در نظر گرفته می‌شود. از این حافظه می‌توان برای تصحیح موقعیت اشیای خارج شده از فضای جست‌وجو و راهنمایی اشیای فعلی استفاده نمود.



شکل (۵) - فلوجارت الگوریتم TEO

^۱ Environmental Objects

^۲ Thermal Objects

۲-۱-۶- الگوریتم سیستم ذرات ارتعاشی

الگوریتم سیستم ذرات ارتعاشی (VPS) یک الگوریتم فراابتکاری توسعه داده شده توسط کاوه و ایلچی‌غازان است [۲۳]. الگوریتم VPS از ارتعاش آزاد سیستم‌های دارای یک درجه‌ی آزادی با میرایی ویسکوز الهام گرفته شده است. الگوریتم VPS با مجموعه‌ای از ذرات ارتعاشی که به طور تصادفی از فضای جست‌وجو انتخاب شده است شروع به کار می‌کند. تعداد ذرات ارتعاشی با nVP نشان داده می‌شود. این ذرات، ماتریس ذرات ارتعاشی (VP) را تشکیل داده که درایه‌های آن شامل پیشنهادهاى الگوریتم VPS برای متغیرهای مجهول تابع هدف است. پس از ارزیابی ذرات در تابع هدف مربوطه، مقادیر تابع هدف جریمه شده تولید می‌شود. الگوریتم VPS ذرات را به گونه‌ای به‌روزرسانی کرده که برای هر ذره سه موقعیت تعادل با وزن‌های مختلف ($\omega_1, \omega_2, \omega_3$) ایجاد می‌شود. این سه موقعیت شامل بهترین موقعیتی که تا کنون در کل جمعیت به دست آمده (HP)، یک ذره‌ی خوب^۱ (GP) و یک ذره‌ی بد^۲ (BP) است. به منظور انتخاب GP و BP برای هر ذره، جمعیت فعلی بر اساس مقادیر تابع هدف جریمه شده‌ی آنها به ترتیب صعودی مرتب شده و سپس GP و BP به ترتیب از نیمه‌ی اول و دوم (به جز خود ذره) به طور تصادفی انتخاب می‌شود.

سطح میرایی^۳ نقش مهمی در ارتعاش دارد. اگر سطح میرایی بالا باشد میزان کاهش دامنه‌ی ارتعاش نیز در واحد زمان بیش‌تر است. به منظور مدل‌سازی این پدیده در VPS یک تابع نزولی (D) متناسب با تعداد تکرارها به صورت زیر پیشنهاد شده است.

$$D = \left(\frac{NITs}{maxNITs} \right)^{-\alpha} \quad (14)$$

در این رابطه NIT تعداد تکرار فعلی الگوریتم، $maxNITs$ حداکثر تکرارهای مجاز برای الگوریتم (معیار توقف) و α یک ثابت است که معمولاً مقدار ۰/۰۵ برای آن توصیه می‌شود. با توجه به مفاهیم ذکر شده، ذرات با رابطه‌ی زیر که رابطه‌ی ارتعاش آزاد نیز نام دارد به‌روزرسانی می‌شوند.

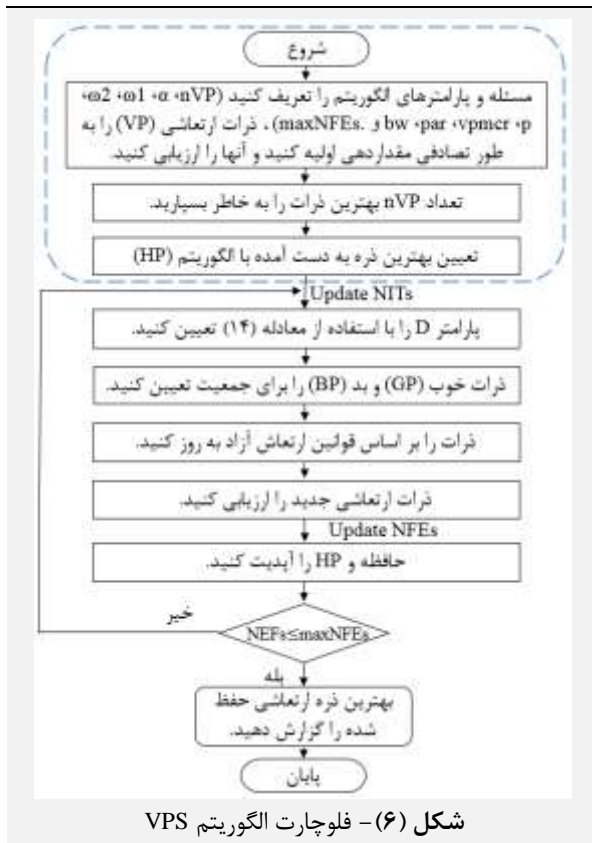
$$newVP_i = \omega_1(D \cdot A \cdot rand + HP) + \omega_2(D \cdot A \cdot rand + GP_i) + \omega_3(D \cdot A \cdot rand + BP_i)$$

$$A = \omega_1(HP - VP_i) + \omega_2(GP_i - VP_i) + \omega_3(BP_i - VP_i) \quad (15)$$

$$\omega_1 + \omega_2 + \omega_3 = 1 \quad (16)$$

در این رابطه VP_i و $newVP_i$ به ترتیب موقعیت فعلی و به‌روزرسانی شده‌ی ذره‌ی i -ام، ω_1, ω_2 و ω_3 سه وزن برای اندازه‌گیری اهمیت نسبی بهترین ذره‌ی تا کنون یافته شده

توسط الگوریتم (HP) و $rand$ -ها اعداد تصادفی بوده که به طور یک‌نواخت بین صفر و ۱ انتخاب می‌شوند. یک پارامتر کنترلی p برای هر ذره تعریف شده که این موضوع که آیا تاثیر BP باید در موقعیت به‌روزرسانی در نظر گرفته شود یا خیر را تعیین می‌کند. برای هر ذره، p با یک عدد تصادفی که از توزیع یک‌نواخت در محدوده‌ی [۰،۱] انتخاب شده است مقایسه می‌شود. اگر $p < rand$ باشد آن‌گاه $\omega_3=0$ و $\omega_2=1-\omega_1$ برقرار است. سه مفهوم اساسی شامل خودسازگاری، هم‌کاری و رقابت در VPS مطرح است. خودسازگاری یعنی یک ذره می‌تواند به سمت HP حرکت کند. هم‌کاری بین ذرات یعنی هر ذره این شانس را دارد که بر موقعیت جدید دیگری تاثیر بگذارد. رقابت یعنی تاثیر GP (ذره‌ی خوب) ممکن است بیش‌تر یا کم‌تر از BP (ذره‌ی بد) باشد. رقابت با پارامتر کنترلی p تنظیم می‌شود.



شکل (۶) - فلوجارت الگوریتم VPS

در الگوریتم VPS از یک حافظه (VP-M) برای ذخیره‌ی تعداد nVP از بهترین ذرات ارتعاشی، تابع هدف آن‌ها (Fit-M) و تابع هدف مجازات شده (PFit-M) استفاده می‌شود. در نظر گرفتن حافظه و بهره‌مندی از آن در قالب استراتژی‌های مختلف می‌تواند عمل‌کرد الگوریتم مذکور را بدون افزایش هزینه‌ی محاسباتی بهبود بخشد. لازم به ذکر است که VPS فقط از این حافظه برای تولید مجدد ذرات خارج شده از فضای جست‌وجو

^۱ Damping Level

^۱ Good Particle

^۲ Bad Particle

است که باید توسط امپراتور یکم تصرف شود. تعداد اولیه‌ی مستعمرات (nCOL) هر امپراتوری به شرح زیر است.

$$nCOL(i) = \text{round}(NP(i) \times nCOL) \quad (19)$$

$$nCOL = nC - nE \quad i = 1 \dots nE$$

در این رابطه تابع round وظیفه‌ی گرد کردن را دارد. در مرحله‌ی ابتدایی الگوریتم ICA برای تقسیم مستعمرات برای هر امپراتور، مستعمرات nCOL(i) به طور تصادفی انتخاب شده و به آن اختصاص داده می‌شود تا امپراتوری i-ام را تشکیل دهد.

۲-۷-۱-۲- مرحله‌ی دوم: جنبش مستعمرات

هر مستعمره در صورت بازدید از موقعیت بهتر در فضای جست‌وجو می‌تواند جای امپراتور را بگیرد. حرکت مستعمره‌ی j-ام از امپراتوری i-ام (EC{i}(j)) با یک جهت تصادفی یک‌نواخت به سمت امپراتوری آن‌ها (eC(i)) و یک انحراف تصادفی یک‌نواخت رخ داده که روابط آن به صورت زیر است.

$$direction = \beta \times U(0, (EC\{i\}(j) - eC(i))) \quad (20)$$

$$i = 1 \dots nE \quad j = 1 \dots nCOL(i)$$

$$deviation = U(-\gamma, +\gamma) \quad (21)$$

$$i = 1 \dots nE \quad j = 1 \dots nCOL(i)$$

$$newEC = EC + direction * deviation \quad (22)$$

در این روابط newEC موقعیت جدید مستعمرات، U(a,b) تابع توزیع یک‌نواخت بین a و b، پارامتری با مقدار بیش‌تر از ۱ و γ پارامتری است که انحراف از جهت اصلی را تنظیم می‌کند. در بسیاری از پیاده‌سازی‌ها مقدار حدود ۲ برای β و حدود $\pi/4$ برای γ منجر به هم‌گرایی مستعمرات به امپراتور می‌شود.

۲-۷-۱-۳- مرحله‌ی سوم: به‌روزرسانی امپراتور

اگر موقعیت جدید مستعمره بهتر از موقعیت امپریالیستی آن باشد (با توجه به عمل کرد تابع هدف جریمه شده)، امپریالیست و مستعمره مواضع خود را تغییر داده، مکان جدید با هزینه‌ی کم‌تر به امپراتوری تبدیل شده و در تکرار بعدی، مستعمره‌های دیگر به سمت این موقعیت جدید حرکت می‌کنند.

۲-۷-۱-۴- مرحله‌ی چهارم: رقابت امپریالیستی

رقابت امپریالیستی، دیگر استراتژی مورد استفاده در ICA است. هر امپراتوری سعی می‌کند مستعمرات سایر امپراتوری‌ها را در اختیار گرفته و آن‌ها را کنترل کند. رقابت امپریالیستی به تدریج قدرت امپراتوری‌های ضعیف‌تر را کاهش و قدرت امپراتوری‌های قدرتمندتر را افزایش می‌دهد. رقابت امپریالیستی فقط با انتخاب

استفاده می‌کند. با توجه به این مکانیسم، هر جز از بردار راه حل که کران‌های بالا یا پایین متغیرها را نقض کند می‌تواند از VP-M برای تولید ذره‌ی جدید استفاده نماید.

۲-۷-۱-۲- الگوریتم رقابتی امپریالیستی

الگوریتم رقابتی امپریالیستی (ICA) توسط آتش‌پز گرگری و هم‌کارانش [۲۴-۲۶] ارائه شده که یک الگوریتم بهینه‌سازی با الهام از مسائل اجتماعی و سیاسی است. الگوریتم ICA در چهار مرحله شامل تشکیل امپراتوری‌های اولیه، جنبش مستعمرات، به‌روزرسانی امپراتور و رقابت امپریالیستی انجام می‌شود. به جز مرحله‌ی اول، سه مرحله‌ی دیگر با هر تکرار الگوریتم تغییر می‌کنند. مانند سایر الگوریتم‌های فراابتکاری مبتنی بر جمعیت، ICA با مجموعه‌ای از راه‌حل‌های اولیه‌ی تصادفی یا کشورها شروع می‌شود. تعداد کشورها با nC و ماتریس جمعیت یا ماتریس کشورها با C نشان داده می‌شود. با جایگذاری درایه‌های ماتریس C در تابع هدف، مقادیر تابع هدف جریمه شده برای هر کشور (PFit) محاسبه می‌شود. مراحل موجود در الگوریتم ICA در ادامه توضیح داده شده است.

۲-۷-۱-۲- مرحله‌ی اول: تشکیل امپراتوری‌های اولیه

در این مرحله تعداد مشخصی (nE) از بهترین کشورها (کشورهایی با توابع هدف مجازات کم‌تر) به عنوان امپراتور انتخاب می‌شود. ماتریس امپراتورها، تابع هدف متناظر آن و بردارهای تابع هدف جریمه شده به ترتیب eC، ePFit و eFit نامیده می‌شود. کشورهای باقی‌مانده مستعمرات این امپریالیست‌ها را تشکیل می‌دهند. به منظور تقسیم متناسب مستعمرات بین امپراتورها، هزینه‌ی نرمالیزه شده^۱ (NC) برای یک امپراتور به صورت زیر تعریف می‌شود.

$$NC(i) = ePFit_i - \max(PFit) \quad i = 1 \dots nE \quad (17)$$

در این رابطه تابع max حداکثر مقدار تابع هدف جریمه شده‌ی امپراتورها را در اختیار قرار می‌دهد. با محاسبه‌ی هزینه‌ی نرمال شده‌ی امپراتورها، توان نرمال شده‌ی (NP) هر امپراتور به صورت زیر تعریف می‌شود.

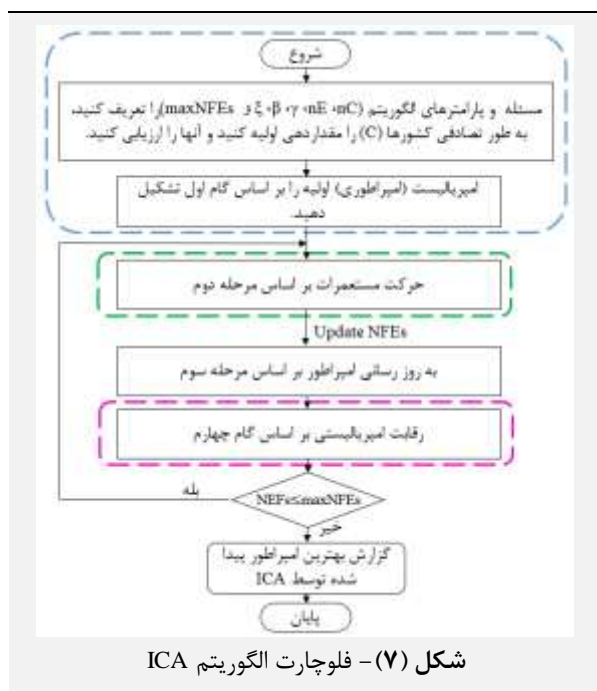
$$NP(i) = \frac{NC(i)}{\sum(NC)} \quad i = 1 \dots nE \quad (18)$$

در این رابطه تابع sum مجموع هزینه‌ی نرمالیزه شده‌ی تمام امپراتورها را محاسبه می‌کند. در واقع NP بخشی از مستعمرات

^۱ Normalized Cost

ترشوندگی سطح باعث می‌شود که تجمع آب روی سطح از حالت لایه‌ای به صورت قطره‌های کوچک میل کند. کاهش ترشوندگی سطح می‌تواند نشان دهنده‌ی کمینه شدن تابع هدف در فرایند بهینه‌سازی باشد. نرخ شار تبخیر مولکول‌های آب به عنوان مناسب‌ترین معیار برای به‌روزرسانی جمعیت در نظر گرفته می‌شود. ابتدا پارامترهای الگوریتم تنظیم شده که این پارامترها تعداد مولکول‌های آب (nWM) و حداکثر تعداد تکرارهای الگوریتم (maxNITs) هستند. الگوریتم WEO با تعداد nWM راه حل (مولکول آب) منتخب از فضای جست‌وجو شروع می‌شود. این راه حل‌ها ماتریس مولکول‌های آب (WM) را می‌سازند. با توجه به متفاوت بودن معادلات فیزیکی مربوط به تبخیر قطرات آب^۲ و مولکول‌های آب روی تک‌لایه^۳، مراحل مختلفی برای ارزیابی و تبخیر مولکول‌های آب در نظر گرفته شده که در ادامه در مورد آن‌ها توضیح داده شده است. هم‌چنین در آغاز الگوریتم WEO پارامترهای دیگری مانند مینیمم و ماکسیمم احتمال تبخیر تک‌لایه (MEP_{min} و MEP_{max}) و مینیمم و ماکسیمم احتمال تبخیر قطرات آب (DEP_{min} و DEP_{max}) باید تعیین شود. پس از ارزیابی مولکول‌ها روی تابع هدف (Fit) مقادیر تابع هدف جریمه شده (PFit) تولید شده و سپس یک ماتریس اندازه‌ی گام مبتنی بر جایگشت تصادفی ایجاد می‌شود. برای مولکول آب i-ام رابطه‌ی زیر برقرار است.

$$stepsize = rand \times (WM[permute1(i)(j)] - WM[permute2(i)(j)]) \quad (27)$$



شکل (۷) - فلوجارت الگوریتم ICA

برخی (معمولاً یکی) از ضعیف‌ترین مستعمرات از ضعیف‌ترین امپراتوری‌ها و ایجاد رقابت بین امپراتوری‌ها برای تصاحب این مستعمرات مدل می‌شود. در این رقابت هر امپراتوری بر اساس قدرت کلی خود امکان تصاحب مستعمرات را خواهد داشت. این واقعیت با تعریف هزینه‌ی کل به صورت زیر مدل می‌شود.

$$TC(i) = ePFit(i) + \xi \times mean(EPFit\{i\}) \quad i = 1 \dots nE \quad (23)$$

در این رابطه TC(i) هزینه‌ی کل امپراتوری i-ام، تابع mean میانگین مقدار توابع هدف جریمه شده‌ی مستعمرات امپراتوری i-ام و ξ عدد مثبتی کم‌تر از ۱ بوده که مقدار ۰/۱ برای آن در اکثر پیاده‌سازی‌ها مقدار مناسبی است. هزینه‌ی کل نرمالیزه شده به صورت زیر تعریف می‌شود.

$$NTC(i) = TC(i) + max(TC) \quad i = 1 \dots nE \quad (24)$$

در این رابطه NTC(i) هزینه‌ی کل نرمالیزه‌ی امپراتوری i-ام است. با داشتن هزینه‌ی کل نرمال شده، احتمال مالکیت هر امپراتوری به صورت زیر محاسبه می‌شود.

$$P(i) = \frac{NTC(i)}{sum(NTC)} \quad i = 1 \dots nE \quad (25)$$

با استفاده از احتمالات محاسبه شده، یک بردار شاخص مالکیت (D) به صورت زیر برای امپراتورها محاسبه می‌گردد.

$$D(i) = P(i) - rand(0.1) \quad i = 1 \dots nE \quad (26)$$

امپراتوری که درایه‌ی مربوطه‌ی آن در D حداکثر باشد ضعیف‌ترین مستعمره را از ضعیف‌ترین امپراتوری تحویل می‌گیرد. نکته‌ی قابل ذکر این است که وقتی یک امپراتوری تمام مستعمرات خود را از دست می‌دهد فرض می‌شود که سقوط کرده است. در این راستا خود امپراتور نیز توسط امپراتوری که حداکثر مقدار را در D دارد تسخیر خواهد شد.

۲-۱-۸- الگوریتم بهینه‌سازی تبخیر آب

در پژوهش کاوه و بخش‌پوری [۲۷] با الهام از تبخیر مقدار کمی از مولکول‌های آب روی سطح جامد با ترشوندگی^۱ الگوریتم فراابتکاری جدیدی به نام بهینه‌سازی تبخیر آب (WEO) توسعه داده شده است. الگوریتم WEO مولکول‌های آب را به عنوان اعضای جمعیت در نظر می‌گیرد. سطح جامد یا بستر با متغیر ترشوندگی به عنوان فضای جست‌وجو تعریف می‌شود. کاهش

^۱ MonoLayer Evaporation

^۱ Wettability

^۲ Droplet Evaporation

در این روابط DEP_{ij} احتمال به‌روزرسانی متغیر J -ام از i -امین مولکول آب بوده و J_0 و P_0 مقادیر ثابتی هستند که معمولاً رابطه‌ی زیر برای آن‌ها برقرار است.

$$J_0 \times P_0 = \frac{1}{26}$$

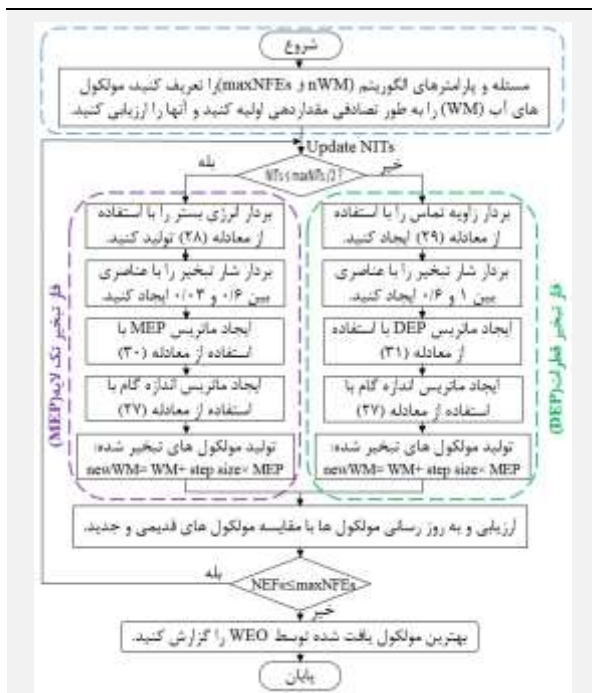
۲-۸-۱-۳- مرحله‌ی به‌روزرسانی مولکول‌های آب

به‌روزرسانی یا تبخیر مولکول‌های آب فعلی با هدف بهبود تابع هدف انجام می‌شود. در WEO تعداد مولکول‌های آب (nWM) در تمام تکرارهای الگوریتم ثابت در نظر گرفته می‌شود. بهترین استراتژی برای بازسازی مولکول‌های آب تبخیر شده، استفاده از مجموعه‌ی فعلی مولکول‌های آب (WM) است. به این ترتیب می‌توان یک اندازه‌ی گام تصادفی مبتنی بر جایگشت برای اصلاح احتمالی در نظر گرفت.

$$stepsize = rand.(WM[permute1(i)(j)] - WM[permute2(i)(j)])$$

مجموعه‌ی بعدی مولکول‌ها ($newWM$) با افزودن این اندازه‌ی گام تصادفی مبتنی بر جایگشت ضرب در احتمال به‌روزرسانی مربوطه (تبخیر تک‌لایه و احتمال تبخیر قطرات) تولید شده و می‌تواند به صورت ریاضی به صورت زیر بیان شود.

$$newWM = WM + stepsize \times \begin{cases} MEP. & NITs < maxNITs/2 \\ DEP. & NITs \geq maxNITs/2 \end{cases} \quad (32)$$



شکل (۸) - فلوجارت الگوریتم WEO

در این رابطه J تعداد متغیرهای مسأله‌ی مورد نظر است. الگوریتم WEO دارای چندین مرحله برای به‌روزرسانی و ارزیابی مولکول‌های آب به شرح زیر است.

۲-۸-۱-۲- مرحله‌ی تبخیر تک‌لایه

در این مرحله انرژی برهم‌کنش زیرلایه^۱ (E_{sub}) درج شده به هر مولکول آب از طریق تابع نرمالیزه‌ی زیر به دست می‌آید.

$$E_{sub}(i) = \frac{(E_{max} - E_{min}) \times (PFit(i) - Min(Fit))}{(Max(Fit) - Min(Fit))} + E_{min} \quad (28)$$

معمولاً در الگوریتم WEO مقادیر -0.5 و $-3/5$ به ترتیب برای E_{min} و E_{max} در نظر گرفته می‌شود. پس از تولید بردار انرژی زیرلایه برای هر مولکول آب، ماتریس احتمال تبخیر تک‌لایه (MEP) با رابطه‌ی زیر ساخته می‌شود.

$$MEP_{ij} = \begin{cases} 1. & \text{if } rand_{ij} < \exp(E_{sub}(i)) \\ 0. & \text{if } rand_{ij} \geq \exp(E_{sub}(i)) \end{cases} \quad (29)$$

در این رابطه MEP_{ij} احتمال به‌روزرسانی متغیر J -ام از i -امین مولکول آب است. به این ترتیب مولکول آب با تابع هدف بهتر احتمال بیش‌تری دارد که در فضای جست‌وجو بدون تغییر باقی بماند. در هر تکرار بدترین و بهترین راه‌حل‌های کاندید شده با احتمال برابر با $MEP_{min} = e^{-3.5} = 0.03$ و $MEP_{max} = e^{-0.5} = 0.6$ به‌روزرسانی می‌شوند.

۲-۸-۱-۲- مرحله‌ی تبخیر قطرات

در این مرحله زاویه‌ی تماس مولکول i -ام $\theta(i)$ به صورت زیر محاسبه می‌شود.

$$\theta(i) = \frac{(\theta_{max} - \theta_{min}) \times (PFit(i) - Min(Fit))}{(Max(Fit) - Min(Fit))} + \theta_{min} \quad (30)$$

در این رابطه θ_{min} برابر با 20 و θ_{max} برابر با 50 درجه‌ی سانتی‌گراد است. پس از محاسبه‌ی زاویه‌ی تماس، شار تبخیر^۲ برای هر مولکول $J(\theta_i)$ و ماتریس احتمال تبخیر قطرات (DEP) با روابط زیر ساخته می‌شود.

$$J(\theta_i) = J_0 P_0 \left(\frac{2}{3} + \frac{\cos \theta_i^3}{3} - \cos \theta_i \right)^{\frac{2}{3}} (1 - \cos \theta_i) \quad (31)$$

$$DEP_{ij} = \begin{cases} 1. & \text{if } rand_{ij} < J(\theta_i) \\ 0. & \text{if } rand_{ij} \geq J(\theta_i) \end{cases}$$

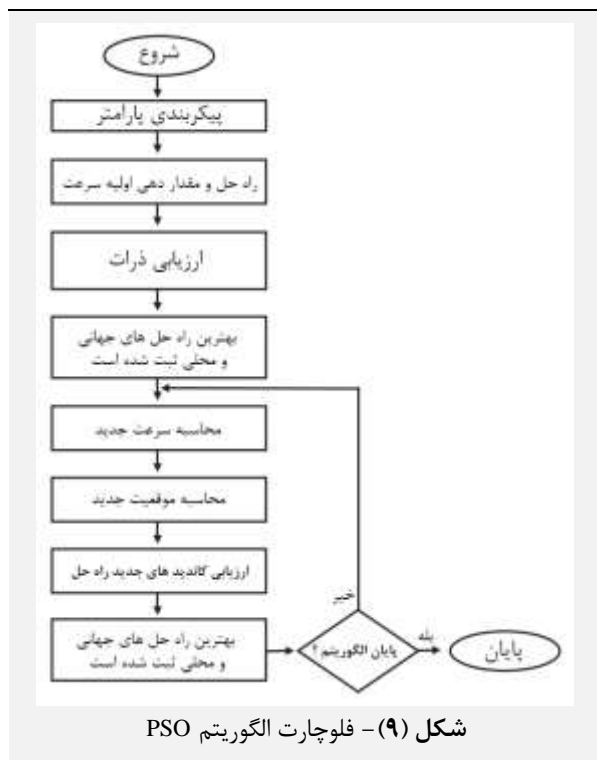
^۱ Evaporation Flux

^۲ Substrate

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (33)$$

$$v_i(t+1) = w V_i(t) + c_1 R_{1i} [P_{best_i}(t) - X_i(t)] + c_2 R_{2i} [G_{best}(t) - X_i(t)] \quad (34)$$

در این روابط t تعداد تکرارهای الگوریتم، w وزن اینرسی، $V_i(t)$ جابه‌جایی با توجه به سرعت فعلی ذره، $X_i(t)$ موقعیت ذره، $P_{best_i}(t)$ جابه‌جایی در جهت بهترین عمل‌کرد ذره‌ی i -ام تا کنون، $G_{best}(t)$ جابه‌جایی در جهت بهترین عمل‌کرد اطلاع‌رسانی شده از ذرات دیگر تا کنون، c_1 ضریب یادگیری شناختی، c_2 ضریب یادگیری اجتماعی و R_1 و R_2 دو عدد تصادفی با توزیع یک‌نواخت در بازه‌ی $[0, 1]$ است. در هر تکرار الگوریتم، ذره‌ای که بهترین جواب سرتاسری^۱ را دارد ثبت شده و در تکرار بعدی، الگوریتم با به‌روزرسانی موقعیت ذرات و مقایسه‌ی جواب‌ها با بهترین جواب یافته شده ادامه می‌یابد تا زمانی که شرط توقف حاصل شود.



۲-۲- پایگاه داده

در این پژوهش از ثبت‌های ECG موجود در پایگاه داده‌ی ریتیم سینوسی نرمال MIT-BIH (NSRDB) استفاده شده است [۱۵]. پایگاه داده‌ی NSRDB شامل ۱۸ ضبط طولانی مدت ECG از افراد مراجعه کننده به آزمایشگاه بیمارستان بث اسرائیل بوستون^۲ است. این افراد هیچ‌گونه آریتمی قابل توجهی نداشته و شامل ۵ مرد ۲۶ تا ۴۵ ساله و ۱۳ زن ۲۰ تا ۵۰ ساله

این مولکول‌های آب جدید بر اساس تابع هدف ارزیابی می‌شوند. اگر تعداد تکرارهای الگوریتم (NIT) از حداکثر تعداد تکرارهای (maxNITs) بیشتر شود الگوریتم خاتمه می‌یابد.

۲-۱-۹- بهینه‌سازی ازدحام ذرات

الگوریتم بهینه‌سازی ازدحام ذرات (PSO) یکی از محبوب‌ترین الگوریتم‌های بهینه‌سازی هوشمند است که در حوزه‌ی الگوریتم‌های مبتنی بر هوش ازدحامی جای می‌گیرد. این الگوریتم توسط کندی و ابرهارت در سال ۱۹۹۵ معرفی شده [۲۸] و با الهام از رفتار اجتماعی حیواناتی چون ماهی‌ها و پرندگان که در گروه‌های کوچک و بزرگ زندگی می‌کنند طراحی شده است. در الگوریتم PSO اعضای جمعیت (جواب‌ها) به صورت مستقیم با هم در ارتباط بوده و از طریق تبادل اطلاعات با یکدیگر و یادآوری جواب‌های خوب گذشته به حل مسائل می‌پردازند. این الگوریتم برای انواع مسائل پیوسته و گسسته مناسب بوده و پاسخ‌های مناسبی را برای انواع مختلفی از مسائل بهینه‌سازی ارائه کرده است. نمایش بلوکی الگوریتم PSO در شکل (۹) قابل مشاهده است. مراحل بهینه‌سازی گام به گام برای الگوریتم PSO به صورت زیر است [۲۹].

۱- تنظیم پارامترهای بهینه‌سازی ازدحام ذرات

۲- ارزیابی تابع برازندگی ذرات

۳- به‌روزرسانی موقعیت و سرعت ذرات

۴- برای هر ذره، تابع برازندگی $x_i(t+1)$ با تابع برازندگی $P_{best_i}(t+1)$ مقایسه می‌شود. اگر مقدار تابع برازندگی این ذره کم‌تر از مقدار تابع برازندگی $P_{best_i}(t+1)$ باشد آن‌گاه $x_i(t+1)$ به عنوان بهترین تجربه‌ی شخصی ذره‌ی i -ام در مرحله‌ی $t+1$ قرار می‌گیرد و در غیر این صورت بهترین تجربه‌ی شخصی ذره‌ی i -ام در مرحله‌ی $t+1$ همان بهترین تجربه‌ی شخصی ذره‌ی i -ام در مرحله‌ی t است.

۵- در مرحله‌ی $t+1$ بهترین تجربه‌ی شخصی ذره‌ای به عنوان $G_{best}(t+1)$ انتخاب می‌شود که بهترین تجربه‌ی شخصی انتخاب شده در بین بهترین تجربه‌های شخصی تمام ذرات، کم‌ترین مقدار تابع برازندگی را داشته باشد.

۶- شرط اتمام الگوریتم بررسی می‌شود. اگر تعداد تکرار الگوریتم به حداکثر تعداد رسیده باشد به عنوان مقدار جواب بهینه یا زیر بهینه اعلام شده و برنامه تمام می‌شود. در غیر این صورت مرحله‌ی ۲ شروع می‌شود.

از نظر ریاضی، موقعیت و سرعت ذرات طبق روابط (۳۳) و (۳۴) به‌روز می‌شود [۳۰].

^۱ Boston's Beth Israel Hospital

^۲ Global Solution

به دنبال مقادیر $[a_j, b_j, \theta_j]^T$ ($i \in \{P, Q, R, S, T\}$) است که تابع زیر را به حداقل می‌رساند.

$$h = \int_{-\pi}^{\pi} \left| \overline{ECG}(\theta) - \sum_{j \in \{P, Q, R, S, T\}} a_j \exp\left(-\frac{(\theta - \theta_j)^2}{2b_j^2}\right) \right|^2 d\theta \quad (36)$$

به صورت نظری، این الگوریتم باید به طور هم‌زمان a_j ، b_j و θ_j مرتبط با شکل موج‌های P ، Q ، R ، S و T را پیدا کند. در واقع یعنی 15 متغیر مجهول را طوری پیدا کند که تابع h کمینه شود. اما تجربه نشان داده که اگر تابع h به شکل رابطه‌ی (۳۶) تعریف شود شکل موج ECG مصنوعی حاصل از پارامترهای یافته شده توسط الگوریتم ممکن است شباهت قابل قبولی به $\overline{ECG}(\theta)$ نداشته باشد. یک دلیل این است که $h(\cdot)$ یک تابع آسان برای بهینه‌سازی نیست زیرا در داخل آن مجموع 5 تابع گوسی غیرخطی وجود دارد. مشکل زمانی بدتر می‌شود که بر اساس پیشنهادهای برخی مراجع فرض شود که شکل موج‌های P و T شکل موج کاملاً گوسی نیست. بنابراین دو یا چند تابع گوسی برای مدل‌سازی هر شکل موج مورد نیاز است. این بدان معنی بوده که حداقل 7 تابع گوسی برای مدل‌سازی $\overline{ECG}(\theta)$ مورد نیاز است که یافتن مقادیر مناسب پارامترهای EDM را برای روش بهینه‌سازی مذکور سخت‌تر می‌کند. از طرف دیگر برای تقریب خوب پارامترها با استفاده از این روش، به نقاطی از سیگنال ECG نیاز است که باید توسط کاربر به صورت دستی انتخاب شود. در صورت انتخاب نادرست این نقاط، عمل کرد روش بهینه‌سازی فوق به شدت تقلیل می‌یابد.

به منظور حل این مشکل‌ها، در این مقاله روش جدیدی برای محاسبه‌ی پارامترهای EDM پیشنهاد شده است. این روش از یک رویکرد تکراری برای یافتن پارامترهای EDM استفاده می‌کند. در هر تکرار، یک تابع متفاوت کمینه می‌شود. اما هر کدام از این توابع بسیار ساده‌تر از تابعی بوده که در رابطه‌ی (۳۶) تعریف شده است زیرا در داخل آن‌ها فقط یک تابع گوسی با پارامترهای مجهول وجود دارد. هم‌چنین دیگر نیازی به انتخاب دستی نقاط در روش پیشنهادی نیست. در روش پیشنهادی در تکرار k -ام، تابع هدف زیر کمینه می‌شود.

$$h_k = \int_{-\pi}^{\pi} \left| \overline{ECG}(\theta) - ECG_{k-1}(\theta) - a_k \exp\left(-\frac{(\theta - \theta_k)^2}{2b_k^2}\right) \right|^2 d\theta \quad (37)$$

بودند. فرکانس نمونه‌برداری در این ثبت‌ها 128 هرتز است. در این مقاله حداقل 20 بخش ECG از قسمت‌های مختلف هر ثبت برای آزمایش‌ها انتخاب شده است. در نتیجه پایگاه داده‌ی آزمایشی در این مقاله شامل 200 سیگنال ECG بوده و طول هر سیگنال حداقل 30 ثانیه است.

۳-۲- روش پیشنهادی

یکی از مدل‌های دینامیکی مورد استفاده در پردازش سیگنال ECG مدل مک‌شری و هم‌کارانش است [۳۱]. در این مدل هر ضربان ECG به صورت ترکیبی از چندین تابع گوسی در نظر گرفته شده است. این مدل هر ضربان ECG را به صورت مجموع 5 تابع گوسی در نظر می‌گیرد که هر کدام از آن‌ها مربوط به موج‌های P ، Q ، R ، S و T است. در این EDM، ضربان‌های ECG در فضای قطبی به دور دایره‌ای با شعاع $r=1$ و فازهای بین 2π (و یا $-\pi$ و π) نگاشت می‌شوند. بسیاری از محققان از این EDM غیرخطی در زمینه‌های مختلف پردازش ECG مانند حذف نویز، بخش‌بندی و تشخیص آریتمی استفاده کرده‌اند [۳۳-۳۲، ۱۲-۱۰، ۳]. این EDM به صورت زیر تعریف می‌شود.

$$ECG = \sum_{j \in \{P, Q, R, S, T\}} a_j \exp\left(-\frac{(\theta - \theta_j)^2}{2b_j^2}\right) \quad (35)$$

در این رابطه a_j ، b_j و θ_j ($j \in \{P, Q, R, S, T\}$) به ترتیب دامنه، گستردگی و مکان قرارگیری توابع گوسی روی دایره در فضای قطبی بوده و رابطه‌ی $-\pi < \theta < \pi$ برقرار است. این مدل غیرخطی بوده و پارامترهای مجهول آن a_j ، b_j و θ_j ($j \in \{P, Q, R, S, T\}$) است. در مقاله‌ی [۱۳] روشی مبتنی بر بهینه‌سازی غیرخطی حداقل مربعات^۱ آفلاین برای پیدا کردن این پارامترهای مجهول پیشنهاد شده است. در این روش ابتدا محل پیک‌های R موجود در هر سیگنال ECG شناسایی شده و فرض می‌شود که این محل‌ها در فضای قطبی در فاز $\theta=0$ قرار می‌گیرند. سپس به نمونه‌های ECG بین دو پیک R متوالی فازهای بین صفر و 2π (یا $-\pi$ و π) تخصیص داده می‌شود. با این کار، هر سیکل ECG روی دایره‌ای فرضی به شعاع $r=1$ در فضای قطبی نگاشت می‌شود. با میانگین‌گیری از نمونه‌هایی که در مکان‌های هم‌فاز قرار گرفته‌اند یک شکل موج میانگین ECG ($\overline{ECG}(\theta)$) ساخته می‌شود. شکل موج میانگین برای هر ECG منحصر به فرد است. پس از محاسبه‌ی میانگین ECG، از یک روش بهینه‌سازی آفلاین برای استخراج پارامترهای ساختمان EDM استفاده می‌شود. به طور دقیق این رویکرد بهینه‌سازی

^۱ Least Square Nonlinear Optimization

پیشنهادی قطعه‌ی S-T و شکل موج T را بسیار بهتر از روش مقاله‌ی [۱۳] مدل‌سازی کرده است. علاوه بر این اندازه‌ی دامنه‌ی شکل موج R در ECG روش پیشنهادی بسیار نزدیک‌تر به دامنه‌ی اصلی شکل موج R در سیگنال ECG میانگین است.

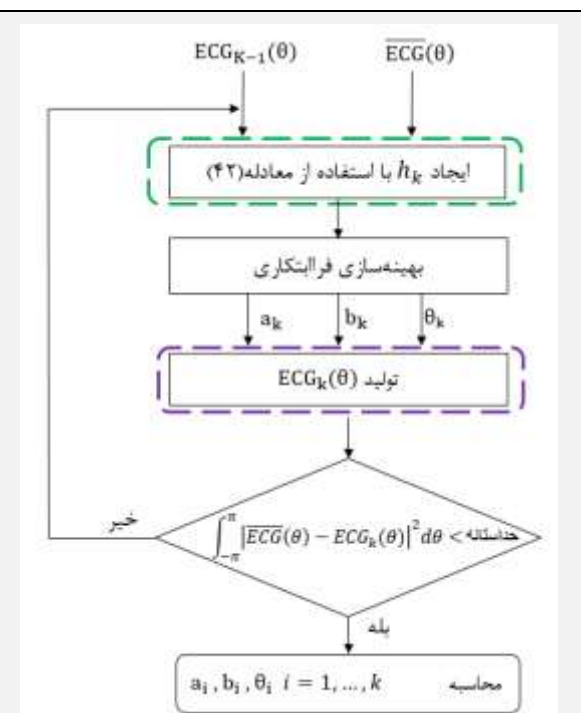
در این رابطه ECG_{k-1} به صورت زیر تعریف می‌شود.

$$ECG_{k-1}(\theta) = \sum_{i=1}^{k-1} a_i \exp\left(-\frac{(\theta - \theta_i)^2}{2b_i^2}\right) \quad (38)$$

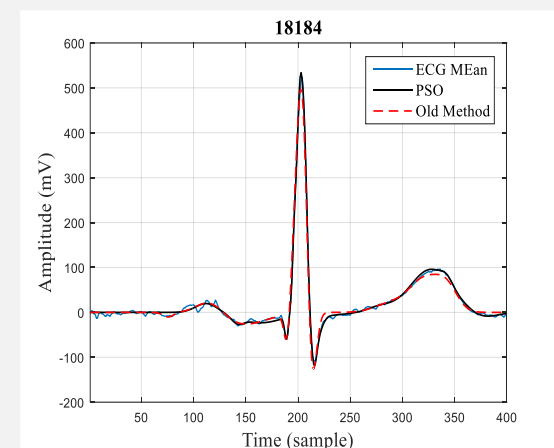
در واقع ECG_{k-1} یک سیگنال مصنوعی بوده که توسط پارامترهای یافته شده تا تکرار $k-1$ ساخته شده است. این روش با $ECG_0(\theta)=0$ شروع شده و تا زمانی ادامه می‌یابد که رابطه‌ی زیر کم‌تر از یک آستانه‌ی کوچک معین باشد.

$$\int_{-\pi}^{\pi} |\overline{ECG}(\theta) - ECG_k(\theta)|^2 d\theta$$

این آستانه با توجه به انرژی کلی $\overline{ECG}(\theta)$ تعریف شده و برای هر ثبت ECG منحصر به فرد است. در این مقاله این آستانه برابر ۱٪ از انرژی کلی $\overline{ECG}(\theta)$ انتخاب شده است. با این تعریف تضمین می‌شود که تقریباً ۹۹٪ از اطلاعات $\overline{ECG}(\theta)$ توسط پارامترهای EDM در ECG_k نهایی حفظ خواهد شد. در این روش برای هر ثبت ECG ممکن است تعداد توابع گوسی مختلفی برای مدل‌سازی پیدا شود که این تعداد به خواص ریخت‌شناسی آن سیگنال و غیرگوسی بودن موج‌های آن بستگی دارد. همچنین مزیت دوم این روش این است که کاملاً خودکار بوده و نیازی به انتخاب دستی نقاط توسط کاربر ندارد. فلوجارت روش پیشنهادی برای استخراج پارامترهای EDM در شکل (۱۰) ارائه شده است. در این مقاله پیشنهاد شده است که



شکل (۱۰) - فلوجارت روش پیشنهادی این مقاله برای استخراج پارامترهای EDM



شکل (۱۱) - مقایسه‌ی روش پیشنهادی (خط تیره) با روش مقاله‌ی [۱۳] (خط چین قرمز)

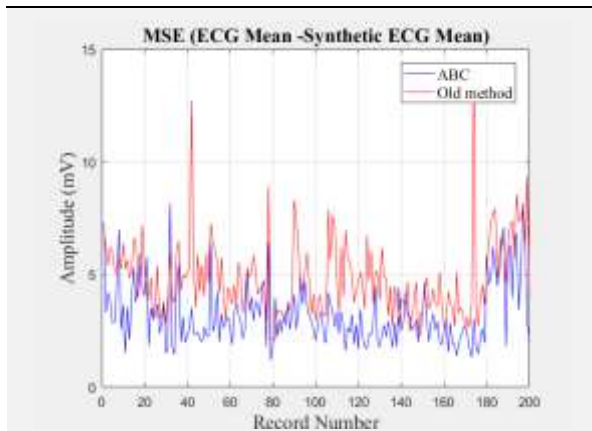
۳- یافته‌ها و بحث

در راستای مقایسه‌ی نتایج هر کدام از الگوریتم‌ها از آن‌جا که از روی نمودارهای ECG تولید شده‌اند، تشخیص این که کدام الگوریتم عمل کرد بهتری داشته سخت است. بدین منظور از دو روش استفاده شده است. در روش اول خطای میانگین مربعات^۱

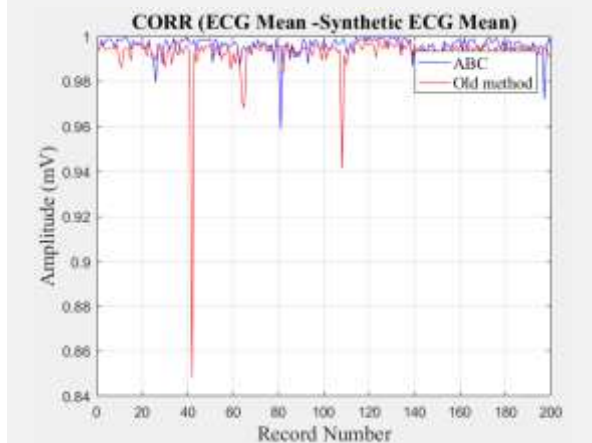
برای بهینه‌سازی تابع موجود در رابطه‌ی (۳۷) از الگوریتم‌های فراابتکاری معرفی شده در بخش قبل به جای رویکرد بهینه‌سازی حداقل مربعات در مقاله‌ی [۱۳] استفاده شود. فرض مقاله این است که روش‌های فراابتکاری به دلیل عدم نیاز به مشتق‌گیری از تابع هدف و کارایی در محیط‌های غیرخطی و فضای جست‌وجوی غیرخطی می‌توانند بهتر از روش بهینه‌سازی مقاله‌ی [۱۳] در مدل‌سازی سیگنال ECG عمل کنند. تفاوت عمل کرد بین روش پیشنهادی و روش قبلی برای یک نمونه‌ی سیگنال میانگین ECG به دست آمده از ثبت ۱۸۱۸۴ از NSRDB در شکل (۱۱) نشان داده شده است [۱۵]. در این شکل از الگوریتم PSO برای بهینه‌سازی استفاده شده است. این شکل دو ECG مصنوعی رسم شده که ECG رسم شده با خط تیره توسط روش پیشنهادی تولید شده و دیگری که با خط چین قرمز ترسیم شده با استفاده از روش مقاله‌ی [۱۳] تولید شده است. در شکل (۱۱) مشاهده می‌شود که روش پیشنهادی در مقایسه با روش قدیمی‌تر، ECG مشابه‌تری به ECG میانگین تولید کرده است. مطابق این شکل روش

^۱ Mean Squared Error

پایین‌تر و NCC بالاتری است. هم‌چنین با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل کرد الگوریتم ABC بهتر از روش بهینه‌سازی غیرخطی پیشین است.



شکل (۱۲) - مقایسه‌ی عمل کرد الگوریتم ABC (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۱۳) - مقایسه‌ی عمل کرد الگوریتم ABC (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

در شکل‌های (۱۴) و (۱۵) مقادیر MSE و NCC مربوط به خروجی روش BB-BC برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم BB-BC دارای MSE پایین‌تر و NCC بالاتری است. با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل کرد الگوریتم BB-BC بهتر از روش بهینه‌سازی غیرخطی پیشین است. مقادیر پارامترهای استفاده شده در الگوریتم BB-BC به صورت $\alpha=1$ و $\beta=0.2$

(MSE) بین ECG-های میانگین ۲۰۰ سیگنال پایگاه داده‌ی آزمایشی و ECG-های سنتز شده توسط الگوریتم‌های فرآبتکاری مقایسه شده است.

$$MSE = \frac{1}{n} \sum_{i=1}^n (ECG_{mean i} - ECG_{synthetic i})^2 \quad (39)$$

در این رابطه ECG_{mean} سیگنال ECG میانگین، $ECG_{synthetic}$ سیگنال ECG سنتز شده توسط الگوریتم‌های فرآبتکاری و n تعداد نمونه‌های ECG_{mean} است. ممکن است ادعا شود که MSE نمی‌تواند تغییرات مورفولوژی سیگنال‌های سنتز شده نسبت به سیگنال‌های ECG میانگین را به طور دقیق مشخص کند. به همین دلیل در روش دوم ضریب همبستگی نرمالیزه شده^۱ (NCC) بین ECG-های میانگین ۲۰۰ سیگنال پایگاه داده‌ی آزمایشی و ECG-های سنتز شده توسط الگوریتم‌های فرآبتکاری محاسبه شده است.

$$NCC(r, l_i, \tau) = \frac{\sum_k ((r(k) - \bar{r}) \times (l_i(k + \tau) - \bar{l}_i))}{\sqrt{\sum_k (r(k) - \bar{r})^2} \sqrt{\sum_k (l_i(k + \tau) - \bar{l}_i)^2}} \quad (40)$$

در این رابطه r سیگنال ECG میانگین، \bar{r} میانگین سیگنال ECG میانگین، l_i سیگنال ECG ساخته شده توسط الگوریتم پیشنهادی، \bar{l}_i میانگین ECG سنتز شده توسط الگوریتم‌های فرآبتکاری و τ میزان شیفت است. در این مقاله از بیشینه‌ی سیگنال NCC برای گزارش نتایج استفاده شده است.

برای مقایسه‌ی عادلانه‌ی روش‌ها، تعداد اعضای جمعیت تمام الگوریتم‌ها و تعداد ماکسیمم تکرار آن‌ها برابر در نظر گرفته شده است. این مقادیر برای تمام الگوریتم‌های فرآبتکاری به ترتیب برابر با ۵۰ و ۲۰۰۰۰ فرض شده است.

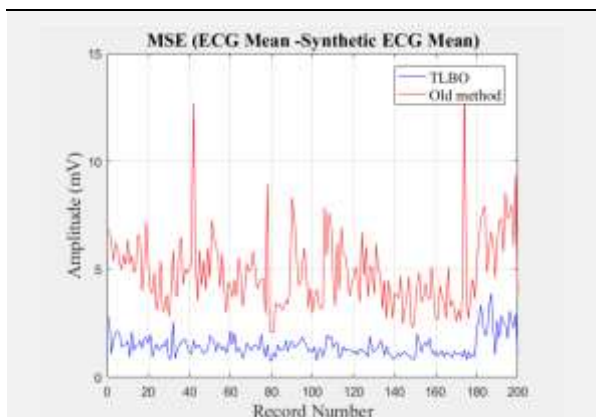
مقادیر MSE و NCC مربوط به خروجی روش‌های فرآبتکاری و خروجی روش مقاله‌ی [۱۳] برای ۲۰۰ سیگنال ECG پایگاه داده‌ی آزمایشی در شکل‌های (۱۲) تا (۲۹) نشان داده شده است. هم‌چنین برای مقایسه‌ی راحت‌تر در جدول (۱) میانگین و انحراف معیار MSE-ها و در جدول (۲) میانگین و انحراف معیار NCC-ها در هر کدام از روش‌ها ارائه شده است.

در شکل‌های (۱۲) و (۱۳) مقادیر MSE و NCC مربوط به خروجی روش ABC برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها، خروجی‌های الگوریتم ABC دارای MSE

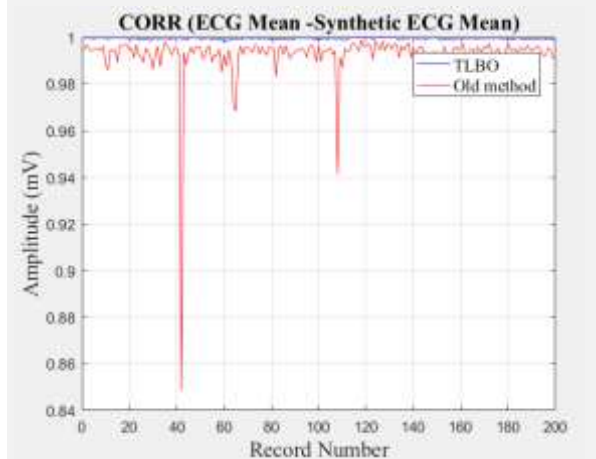
^۱ Normalized Correlation Coefficient

الگوریتم برابر با ۰/۹۹۹۵ بوده که در مقایسه با سایر الگوریتم‌های فراابتکاری یکی از سه الگوریتم برتر پیشنهادی در این مقاله است.

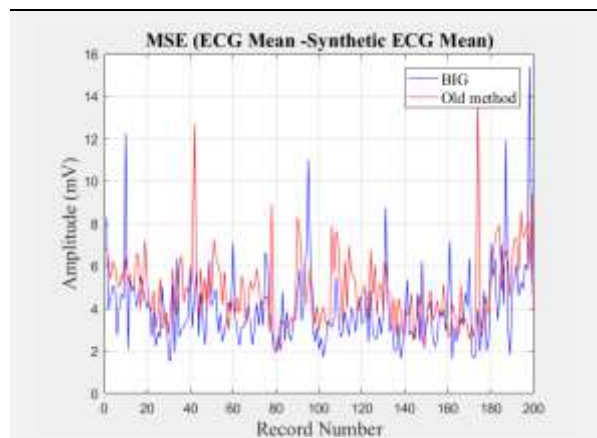
است. در جدول (۱) میانگین MSE الگوریتم BB-BC برابر با ۳/۹۵ بوده که بیش‌تر از الگوریتم ABC است اما هم‌چنان عمل‌کرد آن در مقایسه با روش بهینه‌سازی غیرخطی بهتر است.



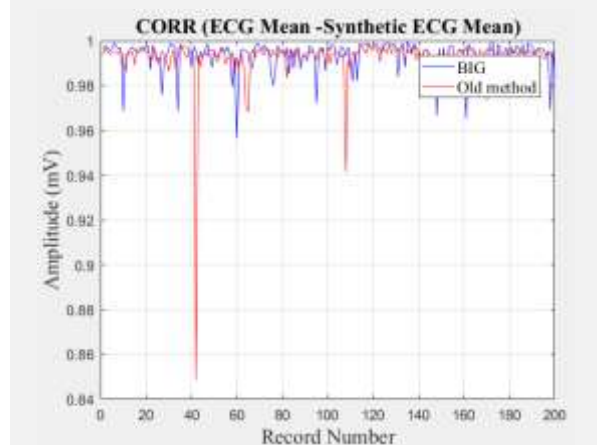
شکل (۱۶) - مقایسه‌ی عمل‌کرد الگوریتم TLBO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۱۷) - مقایسه‌ی عمل‌کرد الگوریتم TLBO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۱۴) - مقایسه‌ی عمل‌کرد الگوریتم BB-BC (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

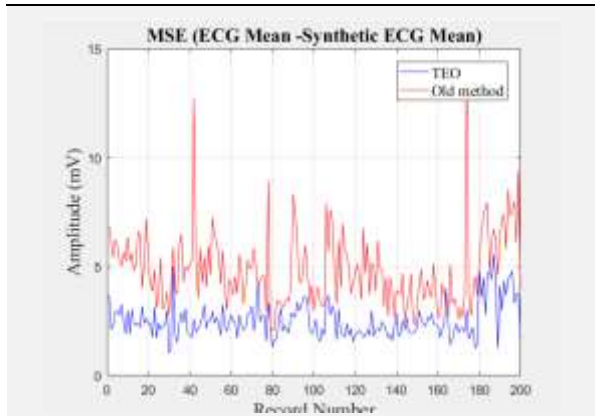


شکل (۱۵) - مقایسه‌ی عمل‌کرد الگوریتم BB-BC (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

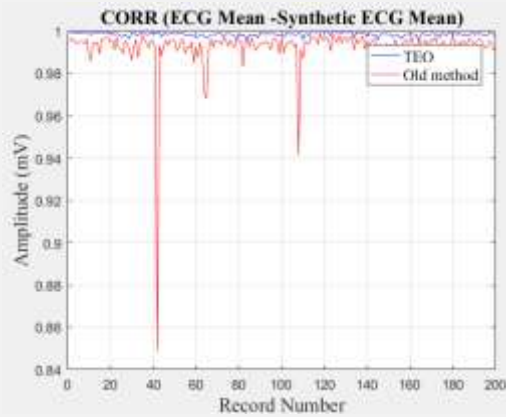
الگوریتم فراابتکاری جست‌وجوی فاخته با میانگین MSE ۱/۵۰ یکی از الگوریتم‌های موفق در این مقاله است. مقادیر پارامترهای الگوریتم CS به صورت $\alpha=1$ و $pa=0.2$ است. در شکل‌های (۱۸) و (۱۹) مقادیر MSE و NCC خروجی روش CS برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی مقاله‌ی [۱۳] مقایسه گردیده است. مطابق این دو شکل در اکثر ثبت‌ها خروجی‌های الگوریتم CS دارای MSE پایین‌تر و NCC بالاتری است. با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل‌کرد الگوریتم CS بهتر از روش بهینه‌سازی غیرخطی پیشین است. میانگین NCC الگوریتم CS برابر با ۰/۹۹۹۵ بوده که مشابه میانگین NCC الگوریتم TLBO است.

در شکل‌های (۱۶) و (۱۷) مقادیر MSE و NCC مربوط به خروجی روش TLBO برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم TLBO دارای MSE پایین‌تر و NCC بالاتری است. هم‌چنین با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل‌کرد الگوریتم TLBO بهتر از روش بهینه‌سازی غیرخطی پیشین است. الگوریتم فراابتکاری مبتنی بر یادگیری و آموزش (TLBO) یکی از دقیق‌ترین الگوریتم‌های استفاده شده در این مقاله بوده و دارای میانگین MSE برابر با ۱/۴۶ است. هم‌چنین میانگین NCC این

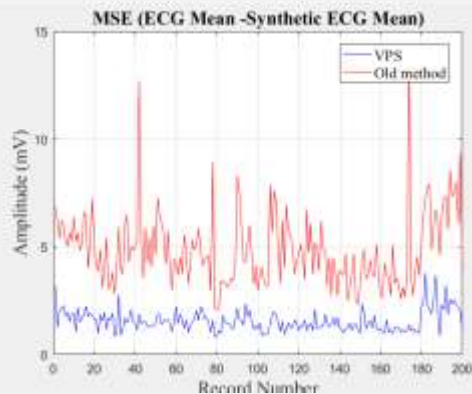
غیرخطی مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم VPS دارای MSE پایین‌تر و NCC بالاتری است. مقادیر اولیه برای این الگوریتم به صورت $w1=0.3$, $\alpha=0.05$, $w2=0.3$, $par=0.1$ و $vpmc=0.95$, $p=0.1$ است.



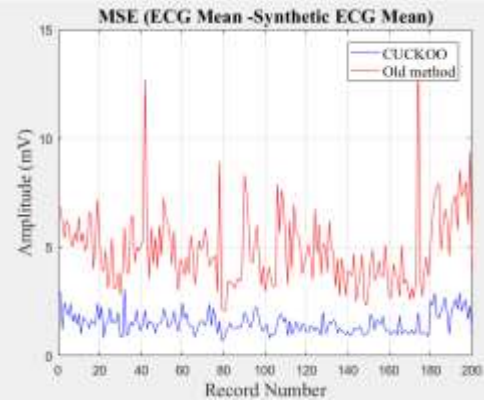
شکل (۲۰) - مقایسه‌ی عمل کرد الگوریتم TEO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



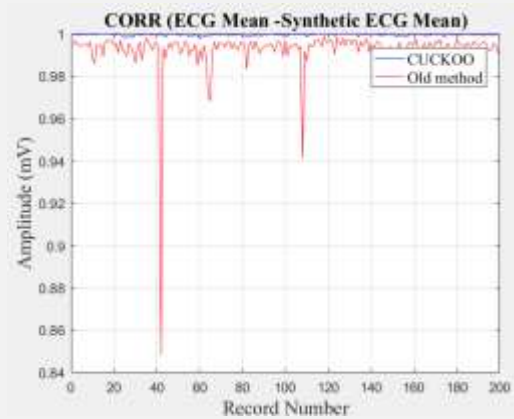
شکل (۲۱) - مقایسه‌ی عمل کرد الگوریتم TEO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۲۲) - مقایسه‌ی عمل کرد الگوریتم VPS (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۱۸) - مقایسه‌ی عمل کرد الگوریتم CS (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



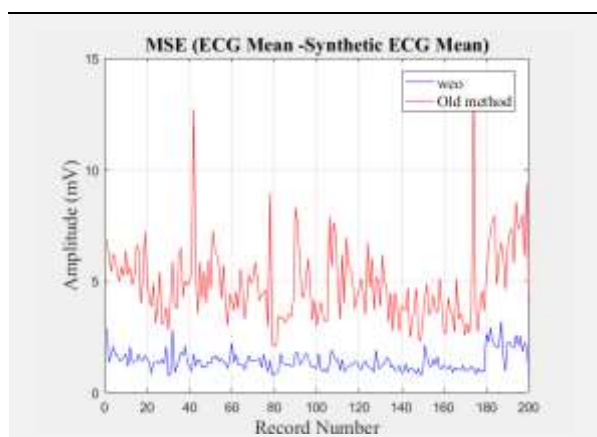
شکل (۱۹) - مقایسه‌ی عمل کرد الگوریتم CS (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

در شکل‌های (۲۰) و (۲۱) مقادیر MSE و NCC مربوط به خروجی روش TEO برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی غیرخطی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم TEO دارای MSE پایین‌تر و NCC بالاتری است. هم‌چنین با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم TEO جداول (۱) و (۲) مشاهده می‌شود که عمل کرد الگوریتم TEO بهتر از روش بهینه‌سازی غیرخطی پیشین است. میانگین MSE و NCC الگوریتم TEO به ترتیب برابر با $2/52$ و 0.9986 است. مقادیر استفاده شده برای پارامترهای الگوریتم TEO نیز به صورت $STO_M=5$ و $p=0.3$ است.

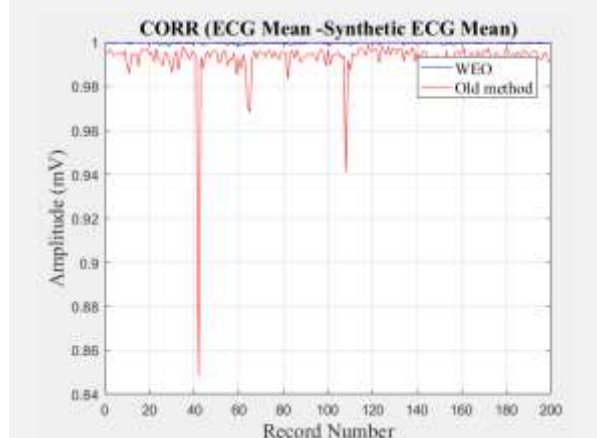
الگوریتم سیستم ذرات ارتعاشی (VPS) با میانگین MSE برابر با $1/55$ و میانگین NCC برابر با 0.9993 از جمله الگوریتم‌های موفق در این مقاله است. در شکل‌های (۲۲) و (۲۳) مقادیر MSE و NCC مربوط به خروجی روش VPS برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی

شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم ICA دارای MSE پایین‌تر و NCC بالاتری است. همچنین با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل‌کرد الگوریتم ICA بهتر از روش بهینه‌سازی غیرخطی پیشین است. میانگین MSE و NCC الگوریتم ICA به ترتیب برابر با ۱/۸۶ و ۰/۹۹۹۱ است. مقادیر استفاده شده برای پارامترهای الگوریتم ICA نیز به صورت $nE=3$ ، $\beta=2$ ، $\gamma=\pi/4$ و $\xi=0/1$ است.

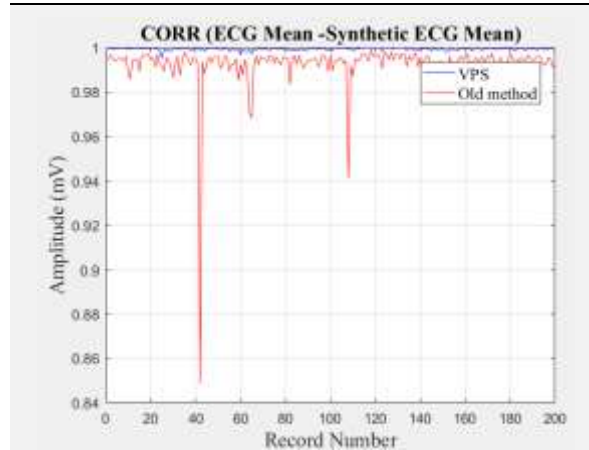
الگوریتم فراباکاری بهینه‌سازی تبخیر آب (WEO) بهترین خروجی را در میان الگوریتم‌های موجود در این مقاله ارائه کرده است. در شکل‌های (۲۶) و (۲۷) مقادیر MSE و NCC مربوط به خروجی روش WEO برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی غیرخطی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. میانگین‌های MSE و NCC این الگوریتم به ترتیب برابر با ۱/۴۰ و ۰/۹۹۹۶ است.



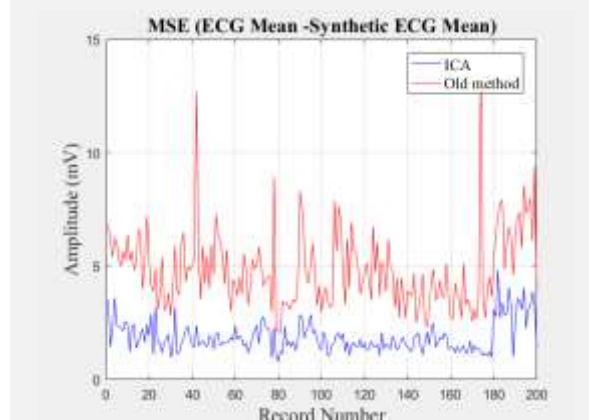
شکل (۲۶) - مقایسه‌ی عمل‌کرد الگوریتم WEO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



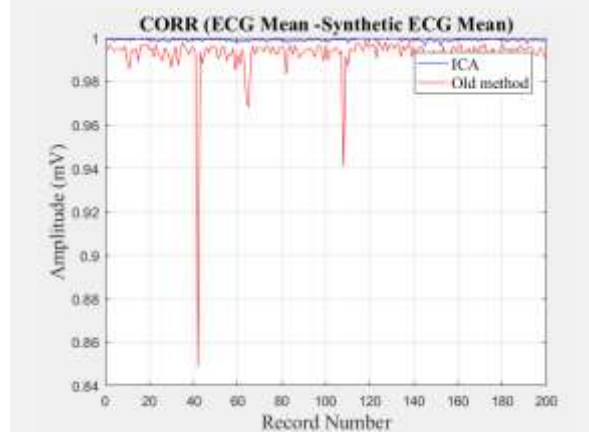
شکل (۲۷) - مقایسه‌ی عمل‌کرد الگوریتم WEO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۲۳) - مقایسه‌ی عمل‌کرد الگوریتم VPS (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۲۴) - مقایسه‌ی عمل‌کرد الگوریتم ICA (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۲۵) - مقایسه‌ی عمل‌کرد الگوریتم ICA (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

در شکل‌های (۲۴) و (۲۵) مقادیر MSE و NCC مربوط به خروجی روش ICA برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی غیرخطی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو

نسبت به سایر الگوریتم‌های فراابتکاری از نگاه دو معیار میانگین MSE و میانگین NCC داشته‌اند.

جدول (۱) - مقایسه‌ی میانگین و انحراف معیار MSE

خروجی‌های روش‌های فراابتکاری

انحراف معیار	میانگین	نام الگوریتم
۱/۷۰۵۸	۴/۸۲	بهینه‌سازی غیرخطی [۱۳]
۱/۴۰۰۳	۳/۲۳	Artificial bee colony (ABC)
۱/۸۷۲۸	۳/۹۵	Big Bang – Big Crunch (BB-BC)
۰/۵۲۰۳	۱/۴۶	Teaching Learning Based Optimization (TLBO)
۰/۴۸۲۷	۱/۵۰	Cuckoo Search (CS)
۰/۶۰۱۰	۱/۷۴	Particle Swarm Optimization (PSO)
۰/۴۴۴۵	۱/۴۰	Water Evaporation Optimization (WEO)
۰/۴۹۷۲	۱/۵۵	Vibrating Particles System (VPS)
۰/۸۰۹۰	۲/۵۲	Thermal Exchange Optimization (TEO)
۰/۶۹۱۸	۱/۸۶	Imperialist Competitive Algorithm (ICA)

جدول (۲) - مقایسه‌ی میانگین و انحراف معیار NCC

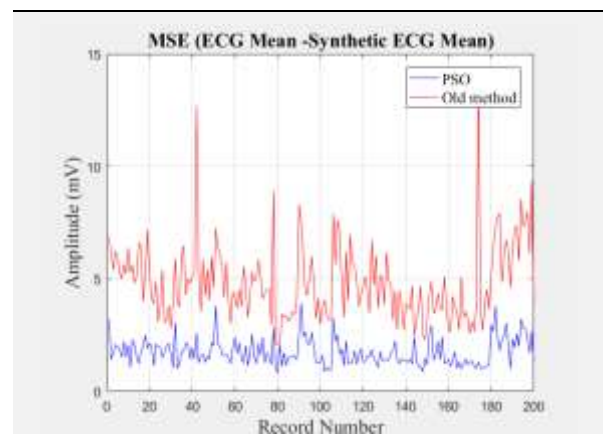
خروجی‌های روش‌های فراابتکاری

انحراف معیار	میانگین	نام الگوریتم
۰/۰۱۱۶۰	۰/۹۹۲۶	بهینه‌سازی غیرخطی [۱۳]
۰/۰۰۴۴۰	۰/۹۹۵۷	Artificial bee colony (ABC)
۰/۰۰۷۱	۰/۹۹۳۲	Big Bang – Big Crunch (BB-BC)
۰/۰۰۰۳۲	۰/۹۹۹۵	Teaching Learning Based Optimization (TLBO)
۰/۰۰۰۳۵	۰/۹۹۹۵	Cuckoo Search (CS)
۰/۰۰۰۵۴	۰/۹۹۹۳	Particle Swarm Optimization (PSO)
۰/۰۰۰۲۷	۰/۹۹۹۶	Water Evaporation Optimization (WEO)
۰/۰۰۰۵۱	۰/۹۹۹۳	Vibrating Particles System (VPS)
۰/۰۰۰۸۴	۰/۹۹۸۶	Thermal Exchange Optimization (TEO)
۰/۰۰۰۵۴	۰/۹۹۹۱	Imperialist Competitive Algorithm (ICA)

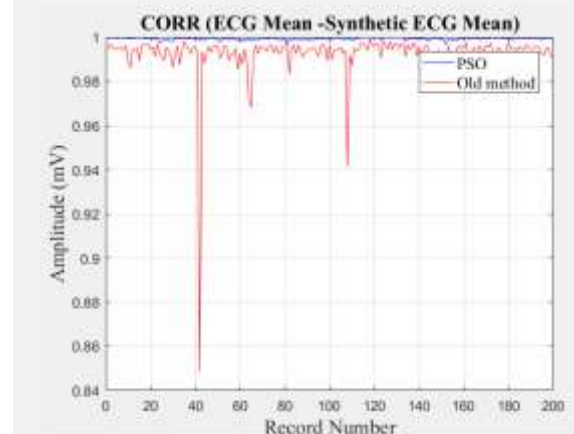
۴- نتیجه‌گیری

در این مقاله عمل کرد تعدادی از الگوریتم‌های فراابتکاری در استخراج مدل دینامیکی سیگنال‌های ECG بررسی شده و مشاهده گردیده که هر ۹ الگوریتم فراابتکاری معرفی شده در این مقاله عمل کرد بهتری نسبت به روش بهینه‌سازی غیرخطی پیشین در استخراج پارامترهای مدل دینامیکی سیگنال‌های ECG داشته‌اند. هم‌چنین به مقایسه‌ی عمل کرد الگوریتم‌های فراابتکاری با یک‌دیگر نیز پرداخته شده و مشاهده گردیده که الگوریتم فراابتکاری WEO بهترین عمل کرد را در این زمینه داشته و پس از آن الگوریتم‌های TLBO و CS به ترتیب دقت بالایی را در استخراج دینامیک سیگنال‌های ECG داشته‌اند.

در شکل‌های (۲۸) و (۲۹) مقادیر MSE و NCC مربوط به خروجی روش PSO برای هر کدام از ۲۰۰ سیگنال آزمایشی رسم شده و با خروجی روش بهینه‌سازی غیرخطی پیشین در مقاله‌ی [۱۳] مقایسه گردیده است. همان‌طور که در این دو شکل مشاهده می‌شود در اکثر ثبت‌ها خروجی‌های الگوریتم PSO دارای MSE پایین‌تر و NCC بالاتری است. هم‌چنین با مقایسه‌ی میانگین‌های MSE و NCC-های دو الگوریتم در جداول (۱) و (۲) مشاهده می‌شود که عمل کرد الگوریتم PSO بهتر از روش بهینه‌سازی غیرخطی پیشین است. میانگین MSE و NCC الگوریتم PSO به ترتیب برابر با ۱/۷۴ و ۰/۹۹۹۳ است. مقادیر استفاده شده برای پارامترهای الگوریتم PSO نیز به صورت $c1=c2=1/49$ است.



شکل (۲۸) - مقایسه‌ی عمل کرد الگوریتم PSO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید MSE روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB



شکل (۲۹) - مقایسه‌ی عمل کرد الگوریتم PSO (خط آبی) با الگوریتم بهینه‌سازی غیرخطی (قرمز) از دید NCC روی ۲۰۰ سیگنال ECG از پایگاه داده‌ی NSRDB

با نگاهی به عمل کرد الگوریتم‌ها در شکل‌های (۱۲) تا (۲۹) و جداول (۱) و (۲) می‌توان نتیجه گرفت که سه الگوریتم فراابتکاری WEO، TLBO و CS به ترتیب عمل کرد بهتری



۵- مراجع

- Physiological measurement, vol. 29, no. 5, pp. 595-613, May 2008, doi: 10.1088/0967-3334/29/5/006.
- [11] O. Sayadi and M. Shamsollahi, "A model-based Bayesian framework for ECG beat segmentation," *Physiological measurement*, vol. 30, no. 3, pp. 335-352, 2009.
- [12] O. Sayadi, M. B. Shamsollahi, and G. D. Clifford, "Robust detection of premature ventricular contractions using a wave-based bayesian framework," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 353-362, 2010.
- [13] G. Clifford, A. Shoeb, P. McSharry, and B. Janz, "Model-based filtering, compression and classification of the ECG," *International Journal of Bioelectromagnetism*, vol. 7, no. 1, pp. 158-161, 2005.
- [14] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67-82, 1997.
- [15] The MIT-BIH Normal Sinus Rhythm Database. PhysioNet, Cambridge, MA [Online] Available: <http://www.physionet.org/physiobank/database/nsrdb/>
- [16] Y. D. Sergeev, D. Kvasov, and M. Mukhametzhanov, "On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget," *Scientific reports*, vol. 8, no. 1, pp. 1-9, 2018.
- [17] K. Iba, "Reactive power optimization by genetic algorithm," *IEEE Transactions on power systems*, vol. 9, no. 2, pp. 685-692, 1994.
- [18] J.-S. Wang and S.-X. Li, "An improved grey wolf optimizer based on differential evolution and elimination mechanism," *Scientific reports*, vol. 9, no. 1, pp. 1-21, 2019.
- [19] O. K. Erol and I. Eksin, "A new optimization method: big bang-big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106-111, 2006.
- [20] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-aided design*, vol. 43, no. 3, pp. 303-315, 2011.
- [21] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.
- [22] A. Kaveh and T. Bakhshpoori, "Metaheuristics: outlines, MATLAB codes and examples," 2019.
- [23] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69-84, 2017.
- [24] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *2007 IEEE congress on evolutionary computation*, 2007: IEEE, pp. 4661-4667.
- [1] M. Akhbari, M. B. Shamsollahi, and C. Jutten, "ECG fiducial points extraction by extended kalman filtering," in *Proc. 36th International Conference on Telecommunications and Signal Processing (TSP)*, 2013: IEEE, 2013, pp. 628-632.
- [2] M. Akhbari, M. B. Shamsollahi, C. Jutten, A. A. Armoundas, and O. Sayadi, "ECG denoising and fiducial point extraction using an extended Kalman filtering framework with linear and nonlinear phase observations," *Physiological measurement*, vol. 37, no. 2, p. 203, 2016.
- [3] M. Akhbari, M. B. Shamsollahi, C. Jutten, and B. Coppa, "ECG denoising using angular velocity as a state and an observation in an extended kalman filter framework," in *Proc. IEEE Annual International Conference on Medicine and Biology Society (EMBC)*, 2012: IEEE, pp. 2897-2900.
- [4] M. M. Hamed Danandeh Hesar, "Implementation of a square-root filtering approach in marginalized particle filters for mixed linear/nonlinear state-space models," *International Journal of Adaptive Control and Signal Processing*, 2019.
- [5] H. Hesar and M. Mohebbi, "ECG Denoising Using Marginalized Particle Extended Kalman Filter with an Automatic Particle Weighting Strategy," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 635-644, 2016, doi: 10.1109/JBHI.2016.2582340.
- [6] H. D. Hesar and M. Mohebbi, "An Adaptive Particle Weighting Strategy for ECG Denoising Using Marginalized Particle Extended Kalman Filter: an Evaluation in Arrhythmia Contexts," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1581-1592, 2017, doi: 10.1109/JBHI.2017.2706298.
- [7] H. D. Hesar and M. Mohebbi, "A Multi Rate Marginalized Particle Extended Kalman Filter for P and T Wave Segmentation in ECG Signals," *IEEE journal of biomedical and health informatics*, vol. 23, no. 1, pp. 112-122, 2018.
- [8] C. Lin, C. Mailhes, and J.-Y. Tournier, "P-and T-wave delineation in ECG signals using a Bayesian approach and a partially collapsed Gibbs sampler," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2840-2849, 2010.
- [9] V. Nathan, I. Akkaya, and R. Jafari, "A particle filter framework for the estimation of heart rate from ECG signals corrupted by motion artifacts," in *Proc. 37th Annual IEEE International Conference on Engineering in Medicine and Biology Society (EMBC)*, 2015: IEEE, pp. 6560-6565.
- [10] R. Sameni, M. B. Shamsollahi, and C. Jutten, "Model-based Bayesian filtering of cardiac contaminants from biomedical recordings,"

- [31] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE transactions on biomedical engineering*, vol. 50, no. 3, pp. 289-294, 2003.
- [32] O. Sayadi and M. B. Shamsollahi, "ECG denoising and compression using a modified extended Kalman filter structure ", *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 9, pp. 2240-2248, 2008.
- [33] M. Akhbari, M. B. Shamsollahi, and C. Jutten, "Fiducial points extraction and characteristic waves detection in ECG signal using a model-based bayesian framework," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013: IEEE, pp. 1257-1261.
- [34] M. Akhbari, M. B. Shamsollahi, and C. Jutten, "Twave alternans detection in ecg using Extended Kalman Filter and dualrate EKF," in *Proc. 22nd European Signal Processing Conference (EUSIPCO)*, 2014: IEEE, pp. 2500-2504.
- [25] E. A. Gargari, F. Hashemzadeh, R. Rajabioun, and C. Lucas, "Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process," *International Journal of Intelligent Computing and Cybernetics*, 2008.
- [26] A. Kaveh and S. Talatahari, "Optimum design of skeletal structures using imperialist competitive algorithm," *Computers & structures*, vol. 88, no. 21-22, pp. 1220-1229, 2010.
- [27] A. Kaveh and T. Bakhshpoori, "Water evaporation optimization: a novel physically inspired optimization algorithm," *Computers & Structures*, vol. 167, pp. 69-85, 2016.
- [28] Y. Shi, "Particle swarm optimization," *IEEE connections*, vol. 2, no. 1, pp. 8-13, 2004.
- [29] P. Umamathy, C. Venkataseshiaiah, and M. S. Arumugam, "Particle swarm optimization with various inertia weight variants for optimal power flow solution," *Discrete Dynamics in Nature and Society*, vol. 2010, 2010.
- [30] Y. He, W. J. Ma, and J .P. Zhang, "The parameters selection of PSO algorithm influencing on performance of fault diagnosis," in *MATEC Web of conferences*, 2016, vol. 63: EDP Sciences, p. 02019.