



دانشگاه صنعتی سهند

شیرین سمانه‌ای غیرخطی در مهندسی برق

دوره ۶ - شماره ۱

بهار و تابستان ۱۳۹۸

صفحات ۴ الی ۳۳

ISSN: 2322-3146

http://jnsee.sut.ac.ir

الگوریتم بهینه‌سازی ازدحام ذرات بهبودیافته مبتنی بر یک کلاس نوین از استراتژی‌های ضرائب یادگیری کارآمد و سریع

محمد جواد عموشاهی^۱، موسی شمسی^۲ و محمدحسین صدیقی^۳

^۱ دانشجوی دکتری، دانشکده مهندسی پزشکی، دانشگاه صنعتی سهند تبریز، m_amoshahy@sut.ac.ir

^۲ نویسنده مسئول، دانشیار، دانشکده مهندسی پزشکی، دانشگاه صنعتی سهند تبریز، shamsi@sut.ac.ir

^۳ استاد، دانشکده مهندسی برق، دانشگاه صنعتی سهند تبریز، sedaaghi@sut.ac.ir

(تاریخ دریافت مقاله: ۱۳۹۷/۰۹/۲۱ تاریخ پذیرش مقاله: ۱۳۹۸/۰۱/۱۳)

چکیده

واژه‌های کلیدی

الگوریتم بهینه‌سازی ازدحام ذرات
بهبودیافته،

ضرائب یادگیری تطبیقی،

ضرائب یادگیری متغیر با زمان،

توابع معیار غیرخطی.

الگوریتم بهینه‌سازی ازدحام ذرات (PSO) یک روش بهینه‌سازی فراابتکاری مبتنی بر جمعیت است که می‌تواند برای طیف وسیعی از مسائل به کار گرفته شود. اما این الگوریتم ایراداتی هم دارد؛ از جمله این که به راحتی در نقاط بهینه محلی گیر می‌افتد و در مراحل پایانی دچار کندی همگرایی می‌شود. به منظور حل این ایرادات، تاکنون انواع الگوریتم‌های PSO بهبودیافته (IPSO) پیشنهاد شده‌اند. به منظور ایجاد یک تعادل بین ویژگی‌های پویا و بهره‌برداری PSO، این مقاله الگوریتم‌های IPSO مبتنی بر یک کلاس جدید از ضرائب یادگیری نمایی (ELF-PSO) را معرفی می‌نماید که از لحاظ محاسباتی کارآمد و سریع هستند. این کلاس شامل استراتژی‌های ضرائب یادگیری نمایی متغیر با زمان (TELF)، ضرائب یادگیری نمایی تصادفی (RELF)، ضرائب یادگیری نمایی خود-تنظیم (SELF) و ضرائب یادگیری نمایی خطی (LELF) است. آزمایش‌های متعددی برای مقایسه روش‌های پیشنهادی با یک مجموعه از استراتژی‌های معروف ضرائب یادگیری ثابت، تصادفی، متغیر با زمان و تطبیقی بر روی یک سری از توابع معیار غیرخطی انجام پذیرفت. نتایج تجربی و تحلیل‌های آماری ثابت می‌کنند که الگوریتم‌های ELF-PSO قادرند دسته وسیعی از مسائل بهینه‌سازی غیرخطی دشوار را به طور کارآمدتری حل کنند. همچنین نتایج آزمایش‌ها نشان می‌دهند که روش‌های پیشنهادی، در اغلب موارد بهتر از سایر الگوریتم‌ها عمل می‌کنند.



Sahand University
of Technology

Journal of Nonlinear
Systems in Electrical
Engineering

Vol.6, No.1

Spring and Summer 2019

ISSN: 2322 – 3146

<http://jnsee.sut.ac.ir>

An Improved Particle Swarm Optimizer based on a Novel Class of Fast and Efficient Learning Factors Strategies

Mohammad Javad Amoshahy¹, Mousa Shamsi² and Mohammad Hossein Sedaaghi³

¹Ph.D. Candidate, Faculty of Biomedical Engineering, Sahand University of Technology, Tabriz, Iran, m_amosahy@sut.ac.ir

²**Corresponding Author**, Associate Prof., Faculty of Biomedical Engineering, Sahand University of Technology, Tabriz, Iran, shamsi@sut.ac.ir

³Prof., Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran, sedaaghi@sut.ac.ir

ABSTRACT

Keywords

Improved particle swarm
optimization algorithm,
Adaptive learning factors,
Time-varying learning
factors,
Nonlinear benchmark
functions.

The particle swarm optimizer (PSO) is a population-based metaheuristic optimization method that can be applied to a wide range of problems. However, it has some drawbacks, e.g., it easily falls into local optima and suffers from slow convergence in the later stages. In order to solve these problems, improved PSO (IPSO) variants have been proposed. To bring about a balance between the exploration and exploitation characteristics of PSO, this paper introduces computationally fast and efficient IPSO algorithms based on a novel class of exponential learning factors (ELF-PSO). This class contains time-varying exponential learning factors (TELF), random exponential learning factors (RELF), self-adjusting exponential learning factors (SELF) and linear-exponential learning factors (LELF) strategies. An experiment is performed and compared with a set of well-known constant, random, time-varying and adaptive learning factors strategies on a suite of nonlinear benchmark functions. The experimental results and statistical analysis prove that the ELF-PSO algorithms are able to solve a wide range of difficult nonlinear optimization problems efficiently. Also, these results show that the proposed methods outperform other algorithms in most cases.

۱- مقدمه

الگوریتم بهینه‌سازی ازدحام ذرات^۱ (PSO) در سال ۱۹۹۵ توسط ابرهارت^۲ و کندی^۳ معرفی شد [۱، ۲]. این الگوریتم، یک روش جستجوی تصادفی^۴ و اجتماعی است که از روی رفتار دسته‌های پرندگان و ماهی‌ها مدل شده است. در PSO هر ذره یک راه‌حل بالقوه برای مسأله را نشان می‌دهد. در این الگوریتم، ذرات در فضای جستجوی مسأله جاری می‌شوند و تغییر مکان ذرات تحت تأثیر تجربه و دانش خودشان و همسایگانشان است. نتیجه مدل‌سازی این رفتار اجتماعی، فرآیند جستجویی است که ذرات به سمت نواحی موفق میل می‌کنند. ذرات از همدیگر می‌آموزند و بر مبنای دانش به‌دست آمده، به سمت بهترین همسایگان خود می‌روند. الگوریتم PSO در سیستم‌های بهینه‌سازی متعددی مورد استفاده قرار می‌گیرد؛ زیرا نیاز به حافظه را کاهش داده است و از نظر محاسباتی دارای کارایی بالایی است؛ به‌علاوه پیاده‌سازی و اجرای آن نسبت به سایر الگوریتم‌های فراابتکاری ساده‌تر است. این الگوریتم به‌طور موفقیت‌آمیزی برای حل تعداد زیادی از مسائل، شامل یافتن نقاط بهینه توابع معیار استاندارد، حل مسائل جایگشتی و آموزش شبکه‌های عصبی به کار گرفته شده است [۳-۵]. یکی دیگر از کاربردهای آن در مهندسی پزشکی برای پردازش تصاویر تشدید مغناطیسی^۵ (MR) است [۶]. بخش‌بندی دقیق تصاویر MR مغز انسان بر اساس نوع بافت یعنی ماده سفید، ماده خاکستری و مایع مغزی-نخاعی، باعث استخراج ویژگی‌هایی می‌شود که به تشخیص بیماری‌ها منتج می‌شود [۷]. این کاربردها نشان می‌دهند که کارایی و دقت الگوریتم PSO حتی در مسائل پزشکی نیز دارای اهمیت حیاتی است [۸].

یکی از نقاط ضعف الگوریتم PSO این است که در شروع اجرا، توانایی جستجوی سراسری بیشتری دارد ولی در مراحل پایانی، توانایی جستجوی محلی افزایش می‌یابد. بنابراین در حل مسائلی که دارای بهینه‌های محلی زیادی هستند، به احتمال زیاد PSO در مراحل پایانی اجرا گرفتار بهینه‌های محلی خواهد شد. البته این احتمال هم وجود دارد که در صورت عدم انتخاب صحیح پارامترهای PSO، این الگوریتم در همان مراحل اولیه اجرا به بهینه‌های محلی همگرا شده و دچار همگرایی پیش‌رس^۶ شود. بنابراین سرعت همگرایی و جلوگیری از افتادن در کمینه‌های محلی دو هدف جذاب در تحقیقات بر روی الگوریتم PSO است و تاکنون تعداد زیادی از نسخه‌های PSO بهبودیافته^۷ (IPSO) برای تحقق این دو هدف ارائه شده‌اند [۹-۱۱]. مسأله مبرهن این است که دسترسی هم‌زمان به دو هدف فوق بسیار مشکل است؛ برای مثال وقتی لیانگ^۸ و همکارانش [۱۲] سعی کردند تا الگوریتم در کمینه‌های محلی گرفتار نشود، دچار کندی همگرایی الگوریتم شدند. به‌منظور فائق آمدن بر مشکلات PSO، همواره تلاش‌هایی صورت گرفته و الگوریتم‌های بهبودیافته‌ای ارائه شده‌اند. برخی از این موارد شامل پیشنهاد وزن اینرسی [۱۳-۱۵]، معرفی PSO مبتنی بر فاکتور انقباض^۹ [۱۶، ۱۷] و پیشنهاد استراتژی‌های ضرائب یادگیری^{۱۰} (LF) [۱۸، ۱۹] هستند.

^۱ Particle Swarm Optimization^۲ Eberhart^۳ Kennedy^۴ Stochastic^۵ Magnetic Resonance^۶ Premature^۷ Improved PSO^۸ Liang^۹ Constriction factor^{۱۰} Learning Factors

موفقیت یک الگوریتم بهینه‌سازی هوشمند مبتنی بر جمعیت در گرو این است که بین قابلیت‌های پویش^۱ و بهره‌برداری^۲ توازن وجود داشته باشد. در الگوریتم PSO، پویش بدین معناست که ذرات، پراکندگی مناسبی داشته باشند تا بتوانند فضای جستجو را مورد کندوکاو قرار دهند و با نمونه‌برداری از نواحی ناشناخته، مناطقی که احتمالاً دارای جواب بهینه هستند را شناسایی کنند. بهره‌برداری به معنای نمونه‌برداری بیشتر از مناطقی است که در بررسی‌های قبلی، نتایج نسبتاً خوبی داشته‌اند و هدف آن، بهبود جواب‌های قبلی است. اگر به قابلیت پویش بهای بیش از حدی داده شود، می‌تواند منجر به سرگردانی ذرات و کندی همگرایی الگوریتم شود. همچنین اگر الگوریتم بیش از اندازه بر قابلیت بهره‌برداری متمرکز شود، می‌تواند منجر به همگرایی ذرات به یک بهینه محلی شود. در این مقاله به منظور ایجاد تعادل بین پویش و بهره‌برداری و افزایش سرعت و دقت الگوریتم PSO، یک کلاس نوین از استراتژی‌های ضرائب یادگیری نمایی^۳ (ELF) پیشنهاد می‌شود که طیف وسیعی از استراتژی‌های ضرائب یادگیری را شامل می‌شود. کلاس استراتژی‌های ELF دارای چهار زیرکلاس است که عبارتند از: زیرکلاس ضرائب یادگیری نمایی متغیر با زمان^۴ (TELF)، زیرکلاس ضرائب یادگیری تصادفی^۵ (RELF)، زیرکلاس ضرائب یادگیری نمایی خود-تنظیم^۶ (SELF) و زیرکلاس ضرائب یادگیری نمایی خطی^۷ (LELF).

طراحی استراتژی‌های ELF به گونه‌ای است که در مراحل آغازین الگوریتم، به تجربیات شخصی ذرات بهای بیشتری داده می‌شود تا ذرات بتوانند فضاهای ناشناخته را کشف کنند. همچنین در مراحل پایانی الگوریتم، بر روی تجربیات جمعی ذرات تأکید بیشتری می‌شود تا سرعت و دقت الگوریتم افزایش یابد. برای اثبات برتری روش‌های پیشنهادی، الگوریتم‌های مبتنی بر استراتژی‌های ELF با یک مجموعه از استراتژی‌های معروف ضرائب یادگیری ثابت، تصادفی، متغیر با زمان و تطبیقی مقایسه خواهند شد. برای این کار، آزمایش‌های متعددی برای حل یک سری از توابع معیار غیرخطی طراحی می‌شوند. با استفاده از نتایج تجربی و تحلیل‌های آماری ثابت می‌شود که الگوریتم‌های پیشنهادی می‌توانند دسته وسیعی از مسائل بهینه‌سازی غیرخطی دشوار را بهتر از سایر الگوریتم‌های مشابه حل کنند.

این مقاله به صورت زیر سازمان‌دهی شده است: در بخش‌های ۲ و ۳ به ترتیب مروری بر الگوریتم PSO و معروف‌ترین استراتژی‌های ضرائب یادگیری ارائه خواهد شد. بخش ۴ به معرفی استراتژی‌های پیشنهادی و الگوریتم‌های جدید IPSO اختصاص دارد. در بخش ۵، شرایط و روش‌های ارزیابی و مقایسه عملکرد الگوریتم‌ها بیان می‌شود. بخش ۶ به نتایج عددی، آزمون‌های آماری و تحلیل نتایج مقایسه الگوریتم‌های پیشنهادی در مقابل الگوریتم‌های موجود می‌پردازد و بهترین الگوریتم را معرفی خواهد کرد. بخش ۷ نیز به نتیجه‌گیری کلی اختصاص یافته است.

۲- مروری بر الگوریتم بهینه‌سازی ازدحام ذرات و پارامترهای آن

در این مقاله منظور از یک مسأله بهینه‌سازی نامقید، یک مسأله کمینه‌سازی^۸ پیوسته است که به صورت زیر تعریف می‌شود:

^۱ Exploration

^۲ Exploitation

^۳ Exponential Learning Factors

^۴ Time-varying Exponential Learning Factors

^۵ Random Exponential Learning Factors

^۶ Self-adjusting Exponential Learning Factors

^۷ Linear-Exponential Learning Factors

^۸ Minimization Problem

تعریف. تابع هدف پیوسته و حقیقی $f(\cdot)$ و فضای جستجوی $S = \{(x_1, x_2, \dots, x_D) : l_d \leq x_d \leq u_d\} \subseteq \mathbb{R}^D$ مفروض است که l_d و u_d به ترتیب کران‌های پایین و بالای بعد d ام ($1 \leq d \leq D$) از S هستند. منظور از کمینه‌سازی تابع $f(\cdot)$ ، یافتن نقطه $X^* \in S$ است به طوری که $f(X^*) \leq f(X)$, $\forall X \in S$.

به منظور حل یک مسأله کمینه‌سازی پیوسته نامقید با تابع هدف $f(\cdot)$ روی فضای جستجوی S ، می‌توان از الگوریتم بهینه‌سازی ازدحام ذرات استفاده کرد.

۲-۱- الگوریتم بهینه‌سازی ازدحام ذرات

در الگوریتم PSO، به صورت تصادفی موقعیت و سرعت قابل قبولی برای ذرات تعیین می‌شوند. موقعیت و سرعت i امین ذره در مرحله t به ترتیب به صورت $X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$ و $V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$ نشان داده می‌شوند که $x_{id}(t) \in [l_d, u_d]$, $d \in [1, D]$ در هر مرحله از الگوریتم PSO استاندارد^۱ (SPSO)، بردارهای موقعیت و سرعت ذرات بر اساس روابط زیر به روز می‌شوند [۲۰]:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1ij} [p_{ij}(t) - x_{ij}(t)] + c_2 r_{2ij} [g_j(t) - x_{ij}(t)] \quad (۱)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \quad (۲)$$

که ω وزن اینرسی^۲ (IW) است و r_{1ij} و r_{2ij} دو عدد مجزای تصادفی با توزیع یکنواخت روی $[0,1]$ هستند. ضرائب شتاب c_1 و c_2 دو عدد ثابت مثبت هستند که به ضرائب یادگیری^۳ (LF) مشهورند و به ترتیب ضرائب یادگیری شناختی^۴ و ضرائب یادگیری اجتماعی^۵ نام دارند [۱] و t نمایش گر شماره تکرار الگوریتم است. بهترین موقعیت قبلی که ذره i ام تاکنون تجربه کرده است را با $P_{besti}(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{iD}(t))$ نمایش می‌دهند و بهترین تجربه شخصی ذره i ام می‌نامند. همچنین بهترین موقعیتی که همه ذرات تاکنون تجربه کرده‌اند را با $G_{best}(t) = (g_1(t), g_2(t), \dots, g_D(t))$ نشان می‌دهند و بهترین تجربه جمعی می‌نامند.

مراحل روش SPSO را می‌توان به صورت زیر بیان نمود:

۱- فرض کنید $t = 0$. یک جمعیت اولیه با ازدحام M و با سرعت‌های متناظر با آن، به صورت تصادفی ایجاد کنید. برای هر i ($1 \leq i \leq M$)، قرار دهید $P_{besti}(0) = X_i(0)$ سپس مقدار $G_{best}(0)$ را با استفاده از رابطه (۳) محاسبه کنید.

$$G_{best}(0) = \arg \min_{X_i(0), 1 \leq i \leq M} f(X_i(0)) \quad (۳)$$

۲- طبق روابط (۱) و (۲) مقدار سرعت و موقعیت ذرات را به روز کنید.

^۱ Standard Particle Swarm Optimization

^۲ Inertia Weight

^۳ Learning Factors

^۴ Cognitive

^۵ Social

۳- طبق رابطه (۴)، برای هر $(1 \leq i \leq M)$ مقدار تابع هدف ذره $X_i(t+1)$ را با مقدار تابع هدف $P_{besti}(t)$ مقایسه کنید. اگر مقدار تابع هدف این ذره کمتر از مقدار تابع هدف $P_{besti}(t)$ است، آن‌گاه $X_i(t+1)$ را به‌عنوان بهترین تجربه شخصی ذره i ام در مرحله $t+1$ قرار بدهید؛ درغیراین صورت بهترین تجربه شخصی ذره i ام در مرحله $t+1$ همان بهترین تجربه شخصی ذره i ام در مرحله t است.

$$P_{besti}(t+1) = \begin{cases} X_i(t+1), & f(X_i(t+1)) < f(P_{besti}(t)) \text{ and } X_i(t+1) \in S \\ P_{besti}(t), & \text{otherwise.} \end{cases} \quad (۴)$$

۴- طبق رابطه (۵)، در مرحله $t+1$ ، بهترین تجربه شخصی ذره‌ای را به‌عنوان $G_{best}(t+1)$ برگزینید که بهترین تجربه شخصی انتخاب شده، در بین بهترین تجربه‌های شخصی همه ذرات، کمترین مقدار تابع هدف $f(\cdot)$ را داشته باشد.

$$G_{best}(t+1) = \arg \min_{P_{besti}(t+1), 1 \leq i \leq M} f(P_{besti}(t+1)) \quad (۵)$$

۵- شرط اتمام الگوریتم را بررسی کنید. اگر تعداد تکرار الگوریتم به حداکثر تعداد تکرار I_{max} رسیده است، $f(G_{best}(t+1))$ را به‌عنوان جواب بهینه یا زیربهینه اعلام کنید و از برنامه خارج شوید؛ درغیراین صورت قرار دهید $t \leftarrow t+1$ و به مرحله ۲ بروید.

۲-۲- پارامترهای الگوریتم PSO

به‌طور کلاسیک، فضای جستجوی الگوریتم PSO به‌صورت یک ابرمکعب $[x_{min}, x_{max}]^D$ در نظر گرفته می‌شود و بردار سرعت هر ذره، در هر مرحله باید در محدودیت $[v_{min}, v_{max}]^D$ صدق کند. شی^۱ و ابرهارت برای $SPSO$ ، وزن اینرسی ثابت (CIW) را معرفی کردند که به‌صورت $\omega = c$ تعریف می‌شود [۲۰]. همچنین آن‌ها بیان کردند که می‌توان از وزن اینرسی متغیر استفاده کرد. با تغییر وزن اینرسی، قابلیت‌های جستجو به‌صورت پویا تنظیم می‌شوند [۲۰]. در [۱۷] یک IW نزولی خطی^۳ $(LDIW)$ معرفی شد و نشان داده شد که در بهبود تنظیم دقیق ویژگی‌های الگوریتم $SPSO$ مؤثر است. در این روش، مقدار ω بر اساس معادله $\omega(t) = \omega_{max} - t \times \frac{\omega_{max} - \omega_{min}}{I_{max}}$ به‌صورت خطی، از مقدار اولیه ω_{max} به مقدار نهایی ω_{min} کاهش می‌یابد که t و I_{max} به‌ترتیب شماره تکرار فعلی و حداکثر تعداد تکرار الگوریتم هستند. این استراتژی بسیار رایج است و الگوریتم‌های زیادی از این روش برای تنظیم IW استفاده می‌کنند. لی^۴ و همکارش [۲۱] یک نوع IW نزولی نمایی^۵ $(EDIW)$ با رابطه $\omega(t) = (\omega_{max} - \omega_{min} - d_1)e^{\frac{I_{max}}{I_{max}+d_2}t}$ معرفی کردند. آزمایش‌های سنگین نشان می‌دهد که عملکرد الگوریتم $SPSO$ با استراتژی $EDIW$ می‌تواند تا حد زیادی بهبود پیدا کند؛ به‌شرطی که $\omega_{max} = 0.95$ ، $\omega_{min} = 0.4$ ، $d_1 = 0.2$ و $d_2 = 7$.

۳- معروف‌ترین استراتژی‌های ضرائب یادگیری

به‌منظور ایجاد تعادل بین قابلیت‌های پویا و بهره‌برداری در الگوریتم PSO، برخی از پژوهشگران بر روی تنظیم ضرائب یادگیری متمرکز شده‌اند و استراتژی‌هایی برای تعیین مقادیر LF پیشنهاد داده‌اند که در این بخش مروری بر این ایده‌ها خواهیم داشت.

^۱ Shi

^۲ Constant Inertia Weight

^۳ Linear Decreasing Inertia Weight

^۴ Li

^۵ Exponent Decreasing Inertia Weight

۳-۱- کلاس LF اولیه

کلاس LF اولیه شامل استراتژی‌هایی می‌شود که مقادیر LF در طول جستجو ثابت یا تصادفی است و هیچ کدام از این روش‌ها نیاز به ورودی ندارند. کندی و ابرهات [۲] یک استراتژی ضرائب یادگیری ثابت^۱ (CLF) به صورت $c_1 = c_2 = 2$ پیشنهاد دادند. کلرک^۲ و کندی [۲۲، ۱۶] یک استراتژی LF مبتنی بر ضریب انقباض^۳ (CFLF) پیشنهاد دادند که سرعت هر ذره بر اساس رابطه $v_{ij}(t+1) = \chi(v_{ij}(t) + \phi_1 r_{1ij}[p_{ij}(t) - x_{ij}(t)] + \phi_2 r_{2ij}[g_j(t) - x_{ij}(t)])$ بر اساس نتایج تجربی، برای CFLF تنظیمات $\omega = \chi = 0.7298$ و $\phi_1 = \phi_2 = 2.05$ پیشنهاد شده‌اند. بنابراین ضرائب یادگیری در این روش به صورت $c_1 = c_2 = \chi\phi_1 = \chi\phi_2 = 1.4962$ هستند. خان^۴ و همکارانش [۱۹] یک استراتژی ضرائب یادگیری تصادفی^۵ (RLF) برای PSO معرفی کردند که از قواعد تصادفی $c_1 = 1 + \text{Rand}()$ و $c_2 = e - \text{Rand}()$ استفاده می‌کند که $\text{Rand}()$ یک عدد تصادفی با توزیع یکنواخت در بازه $[0,1]$ است. همچنین آن‌ها یک استراتژی IW تصادفی^۶ (RIW) با قاعده $\omega = \frac{\text{Rand}()}{M}$ معرفی کردند. در روش RIW هر ذره از یک IW متفاوت استفاده می‌کند تا قابلیت بهره‌برداری را برای ذرات مهیا کند.

۳-۲- کلاس LF متغیر با زمان

استراتژی‌های کلاس LF متغیر با زمان، روش‌هایی هستند که دارای دو تابع بر حسب زمان یا شماره تکرار الگوریتم هستند. شوگانتان^۷ [۲۳] یک استراتژی LF متغیر با زمان با رابطه $c_1 = c_2 = \frac{(c_{\text{upper}} - c_{\text{lower}}) \times (I_{\text{max}} - t)}{I_{\text{max}}} + c_{\text{lower}}$ پیشنهاد داد که در این روش، مقادیر c_1 و c_2 به صورت خطی از یک کران بالا c_{upper} به یک کران پایین c_{lower} کاهش می‌یابند. رانتاویرا^۸ و همکارانش [۲۴] یک تکنیک انتخاب LF را مورد بررسی قرار دادند که استراتژی ضرائب شتاب متغیر با زمان^۹ (TVAC) نامیده می‌شود. در استراتژی TVAC ضرائب یادگیری با استفاده از روابط $c_1(t) = \frac{(c_{1f} - c_{1i})t}{I_{\text{max}}} + c_{1i}$ و $c_2(t) = \frac{(c_{2f} - c_{2i})t}{I_{\text{max}}} + c_{2i}$ به دست می‌آیند. در این روش، بهترین مقادیر برای پارامترها به صورت $c_{1i} = c_{2f} = 2.5$ و $c_{1f} = c_{2i} = 0.5$ هستند. آن‌ها پیشنهاد کردند که این روش با یک IW متغیر با زمان مانند استراتژی LDIW اجرا شود.

۳-۳- کلاس LF تطبیق پذیر

کلاس LF تطبیق پذیر شامل روش‌هایی است که از یک یا چند پارامتر بازخورد استفاده می‌کنند. این پارامترها در هر مرحله، موقعیت الگوریتم را بررسی می‌نمایند و بر اساس این بازخورد، مقادیر مناسب را برای LF تنظیم می‌کنند. جدن^{۱۰} و همکارانش [۲۵] نسخه جدیدی از PSO را معرفی کردند که از یک استراتژی به نام ضرائب شتاب خود-تطبیقی^{۱۱} (SAAF) براساس روابط

¹ Constant Learning Factors

² Clerc

³ Constriction Factor based Learning Factors

⁴ Khan

⁵ Random Learning Factors

⁶ Random IW

⁷ Suganthan

⁸ Ratnaweera

⁹ Time-Varying Acceleration Coefficients

¹⁰ Jadon

¹¹ Self-Adaptive Acceleration Factors

ازدحام در مرحله t ام است و $f_{\max}(t)$ به صورت زیر محاسبه می‌شود:

$$\text{Prob}_i(t) = \frac{f(X_i(t))}{f_{\max}(t)} \quad c_2 = 0.5 + \text{Prob}_i(t) \text{ و } c_1 = 1.2 - \text{Prob}_i(t)$$

برازندگی نسبی ذره i ام نسبت به

$$f_{\max}(t) = \max_{X_i(t), 1 \leq i \leq M} f(X_i(t)) \quad (6)$$

آگراوال^۱ و همکارانش [۱۸] یک روش خودکار جدید برای تعیین پارامترهای LF طراحی کردند و آن را استراتژی LF تطبیق‌پذیر^۲ (ALF) نامیدند. در این روش مقادیر LF با استفاده از روابط $c_1(t) = (c_{1f} - c_{1i}) \left(\frac{f_{\max}(t) - f_{\min}(t)}{f_{\max}(t)} \right) + c_{1i}$ و $c_2(t) = (c_{2f} - c_{2i}) \left(\frac{f_{\max}(t) - f_{\min}(t)}{f_{\max}(t)} \right) + c_{2i}$ به دست می‌آیند که مقدار $f_{\max}(t)$ از رابطه (۶) و مقدار $f_{\min}(t)$ از رابطه زیر به دست می‌آید:

$$f_{\min}(t) = \min_{X_i(t), 1 \leq i \leq M} f(X_i(t)) \quad (7)$$

آن‌ها پیشنهاد کردند که این روش با استراتژی LDIW اجرا شده و پارامترها به صورت $c_{1f} = c_{2i} = 0.5$ و $c_{1i} = c_{2f} = 2.5$ تنظیم شوند.

۴- الگوریتم‌های پیشنهادی ELF-PSO

در این بخش ابتدا برای بهبود عملکرد الگوریتم PSO، یک کلاس جدید از ضرائب یادگیری تصادفی، متغیر با زمان و تطبیق‌پذیر پیشنهاد می‌دهیم. سپس پارامترهای چهار الگوریتم PSO بهبود یافته پیشنهادی را تشریح خواهیم کرد.

۴-۱- استراتژی‌های ضرائب یادگیری نمایی

در روش‌های بهینه‌سازی مبتنی بر جمعیت، در طی مراحل اولیه الگوریتم، مطلوب است که ذرات را تشویق کنیم تا در کل فضای جستجو پراکنده شوند، بدون اینکه در اطراف یک بهینه محلی تجمع کنند؛ اما در طی مراحل بعدی الگوریتم، بسیار مهم است که همگرایی به سمت بهینه سراسری افزایش یابد تا راه‌حل بهینه را با کارایی بالا به دست آوریم. به منظور تحقق این اهداف، چهار استراتژی جدید LF تعریف می‌کنیم. در این تعاریف فرض می‌شود که $t \in [0, I_{\max}]$ یک عدد صحیح باشد.

تعریف. یک استراتژی ضرائب یادگیری نمایی متغیر با زمان (TELF) را به صورت زیر تعریف می‌کنیم:

$$c_1(t) = c_{1i} \left(\frac{c_{1f}}{c_{1i}} \right)^{\frac{t}{I_{\max}}}, c_2(t) = c_{2i} \left(\frac{c_{2f}}{c_{2i}} \right)^{\frac{t}{I_{\max}}} \quad (8)$$

و شرایط اولیه به صورت $c_2(I_{\max}) = c_{2f}$ و $c_2(0) = c_{2i}$ ، $c_1(I_{\max}) = c_{1f}$ ، $c_1(0) = c_{1i}$ و شرایط اولیه به صورت $c_1(I_{\max}) = c_{1f}$ ، $c_1(0) = c_{1i}$ و به ترتیب نشان‌دهنده مقادیر ابتدایی و انتهایی LF شناختی بوده و مقادیر مثبت و ثابت c_{2f} و c_{2i} به ترتیب نشان‌دهنده مقادیر ابتدایی و انتهایی LF اجتماعی هستند. الگوریتم PSO مبتنی بر استراتژی TELF را TELF-PSO می‌نامیم. در واقع مقادیر LF که در الگوریتم SPSO ثابت بودند، اکنون در الگوریتم TELF-PSO با توابع متغیر با زمان جایگزین شده‌اند.

¹ Agrawal

² Adaptive Learning Factors

تعریف. یک استراتژی ضرائب یادگیری نمایی تصادفی (RELF) به صورت زیر تعریف می‌شود:

$$c_1(t) = c_{1i} \left(\frac{c_{1f}}{c_{1i}} \right)^{\frac{t}{I_{\max}}} - \frac{\text{Rand}(0)}{2}, c_2(t) = c_{2i} \left(\frac{c_{2f}}{c_{2i}} \right)^{\frac{t}{I_{\max}}} + \frac{\text{Rand}(0)}{2} \quad (9)$$

که $c_{1f} > \frac{1}{2}$ و تابع $\text{Rand}(0)$ در هر بار فراخوانی یک عدد تصادفی جدید در بازه $[0,1]$ تولید می‌کند. الگوریتم PSO مبتنی بر استراتژی RELF را RELF-PSO می‌نامیم.

تعریف. یک استراتژی ضرائب یادگیری خود-تنظیم (SELF) را به صورت زیر تعریف می‌کنیم:

$$c_1(t) = c_{1i} \left(\frac{c_{1f}}{c_{1i}} \right)^{\frac{t}{I_{\max}}} - \gamma(t), c_2(t) = c_{2i} \left(\frac{c_{2f}}{c_{2i}} \right)^{\frac{t}{I_{\max}}} + \gamma(t) \quad (10)$$

$$\gamma(t) = \frac{f_{\max}(t) - f_{\min}(t)}{2f_{\max}(t)} \quad (11)$$

که $c_{1f} > \frac{1}{2}$ و مقادیر $f_{\min}(t)$ و $f_{\max}(t)$ به ترتیب با استفاده از روابط (۶) و (۷) به دست می‌آیند.

استراتژی SELF، ضرائب یادگیری شناختی و اجتماعی را بر اساس مقدار تابع هدف تنظیم می‌کند. در مقایسه با SPSO، ضرائب یادگیری SELF-PSO وابستگی بسیار زیادی به تجارب قبلی ذرات ندارند.

تعریف. یک استراتژی ضرائب یادگیری خطی (LELF) به صورت زیر تعریف می‌شود:

$$c_1(t) = c_{1i} \left(\frac{c_{1f}}{c_{1i}} \right)^{\frac{t}{I_{\max}}}, c_2(t) = \frac{(c_{2f} - c_{2i})t}{I_{\max}} + c_{2i} \quad (12)$$

الگوریتم PSO مبتنی بر استراتژی LELF را LELF-PSO می‌نامیم.

چهار استراتژی پیشنهادی فوق را استراتژی‌های کلاس ضرائب یادگیری نمایی (ELF) نام‌گذاری کرده و الگوریتم‌های PSO که بر اساس کلاس ELF طراحی می‌شوند را ELF-PSO می‌نامیم. می‌دانیم یک LF اجتماعی بزرگ‌تر این امکان را دارد که ذرات را دچار همگرایی پیش‌رس کند و در نتیجه ذرات به یک بهینه محلی همگرا شوند. در مقابل، یک LF شناختی بزرگ‌تر، باعث سرگردانی بیش از حد ذرات خواهد شد. در استراتژی‌های کلاس ELF، تابع $c_1(t)$ چنان طراحی شده است که در مراحل اولیه بهینه‌سازی، مقادیر بزرگی را اختیار کند و تأثیر تجربه شخصی در سرعت ذرات را افزایش داده و به ذرات اجازه دهد تا به جای حرکت به سمت بهترین ذرات، در سرتاسر فضای جستجو حرکت کنند. تابع $c_1(t)$ به مرور زمان کاهش می‌یابد تا مانع از سرگردانی ذرات در مراحل نهایی بهینه‌سازی شود. همچنین تابع $c_2(t)$ به گونه‌ای طراحی شده است تا در مراحل اولیه بهینه‌سازی، مقادیر کوچکی را اختیار کند و تأثیر تجربه جمعی را در سرعت ذرات کاهش دهد و با این استراتژی، مانع از همگرایی پیش‌رس ذرات به بهینه‌های محلی شود. مقدار تابع $c_2(t)$ به مرور زمان بیشتر می‌شود تا تأثیر تجربه جمعی در سرعت ذرات را افزایش داده و در مراحل بعدی بهینه‌سازی به ذرات اجازه دهد که به سمت بهینه سراسری همگرا شوند. به عبارت دیگر، هدف نهایی

استراتژی‌های کلاس ELF این است که بده‌بستان بین پویش و بهره‌برداری را متعادل کنند. بنابراین معادله به‌روزرسانی سرعت در الگوریتم‌های ELF-PSO به صورت زیر است:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1(t)r_{1ij}[p_{ij}(t) - x_{ij}(t)] + c_2(t)r_{2ij}[g_j(t) - x_{ij}(t)] \quad (13)$$

که $c_1(t)$ و $c_2(t)$ به ترتیب ضرائب یادگیری شناختی و اجتماعی هستند و بسته به نوع استراتژی، بر اساس روابط (۸) تا (۱۲) به دست می‌آیند. نمودار استراتژی‌های پیشنهادی SELF، RELF، LELF، TELF در شکل ۱ نمایش داده شده است.

۴-۲- پارامترهای الگوریتم‌های ELF-PSO

به منظور تعیین بهترین مقادیر برای پارامترهای c_{1f} ، c_{1i} ، c_{2f} و c_{2i} شبیه‌سازی‌های متعددی با توابع معیار غیرخطی انجام گرفت. بر اساس مشاهدات، شرط بهبود در جواب بهینه برای اغلب توابع معیار $0 < c_{1i} < 1 < c_{1f} < 3$ و $0 < c_{2i} < 1 < c_{2f} < 3$ است. همان گونه که در جدول ۱ نیز نشان داده شده است، از چهار گونه استراتژی‌های پیشنهادی کلاس ELF برای مقایسه با شش استراتژی LF معروف و پرکاربرد استفاده خواهد شد.

۵- شرایط و روش‌های ارزیابی و مقایسه عملکرد الگوریتم‌ها

برای مقایسه الگوریتم‌ها با یکدیگر می‌بایست شرایط انجام رقابت، تنظیمات پارامترها، معیارهای ارزیابی عملکرد و توابع معیار کاملاً شفاف باشد تا نتایج مقایسه‌ها از قابلیت اطمینان بالایی برخوردار باشند. بنابراین موارد فوق به‌طور جداگانه تشریح خواهند شد و نتایج به دست آمده از الگوریتم‌ها توسط روش‌های آماری، مورد ارزیابی قرار خواهند گرفت تا بهترین الگوریتم مشخص شود.

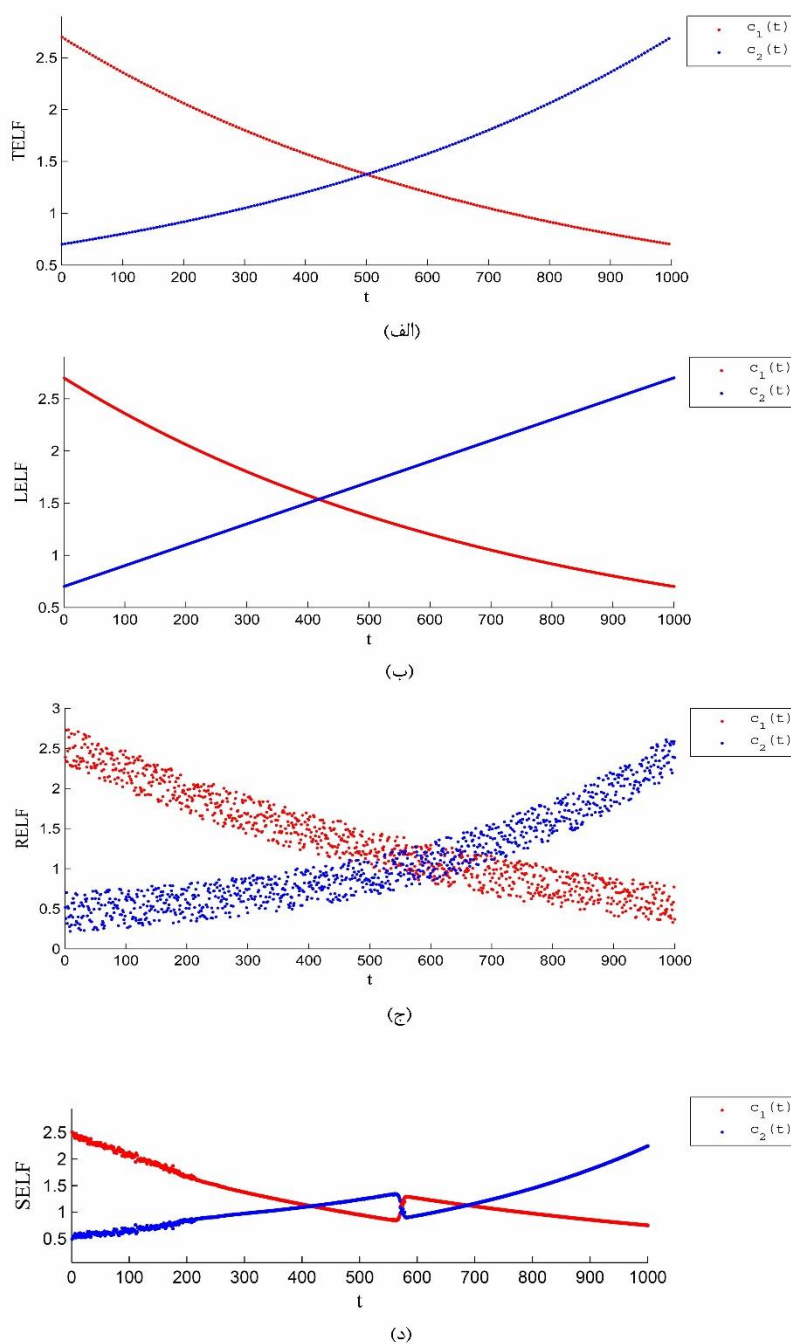
۵-۱- شرایط انجام رقابت و تنظیم پارامترهای الگوریتم‌ها

برای انجام یک رقابت منصفانه، پارامترها و محیط اجرای همه الگوریتم‌ها باید یکسان باشند. به همین دلیل در این بخش شرایط اجرای الگوریتم‌ها را بیان می‌کنیم. بر پایه پژوهش‌های انجام شده، شش LF معروف و پرکاربرد را برای مقایسه با چهار گونه از الگوریتم‌های ELF-PSO انتخاب کردیم که پارامترهای این ده استراتژی در جدول ۱ آمده است. برای انجام مقایسه‌های رقابتی، برنامه مربوط به همه الگوریتم‌ها را با زبان برنامه‌نویسی متلب نسخه ۲۰۱۴ پیاده‌سازی کردیم و تمام الگوریتم‌ها را با سیستم رایانه‌ای یکسان مورد ارزیابی قرار دادیم. همچنین به منظور کاهش اثر تصادفی بودن این الگوریتم‌ها و دستیابی به نتایج دقیق‌تر، هر برنامه ۱۰۰ مرتبه با رابطه محدودسازی سرعت $v_{\max,j} = 0.1 \times (x_{\max,j} - x_{\min,j})$ ، ازدحام ذرات اولیه تصادفی با اندازه $M = 5 \times D$ و بعد توابع معیار $D = 10$ اجرا شد.

۵-۲- معیارهای ارزیابی عملکرد الگوریتم‌ها

بر اساس قضیه «هیچ ناهاری مجانی نیست» [۲۶] یک الگوریتم بهینه‌سازی نمی‌تواند نسبت به بقیه الگوریتم‌ها از هر جهت یا روی هر نوع مسأله‌ای عملکرد بهتری داشته باشد. بنابراین برای سنجش الگوریتم‌ها نیاز به چندین معیار ارزیابی عملکرد (PEC) متفاوت و توابع معیار استاندارد متنوع و معروف است تا نقاط ضعف و قوت الگوریتم‌ها، بهتر نمایان شده و نتایج نهایی معتبر باشند.

¹ No Free Lunch Theorem



شکل ۱. نمودار استراتژی‌های کلاس ضرائب یادگیری نمایی.

(الف) استراتژی TELF. (ب) استراتژی LELF. (ج) استراتژی RELF. (د) استراتژی SELF برای تابع معیار Rastrigin.

تعریف. فرض کنید X^* راه‌حل بهینه سراسری تابع معیار $f(\cdot)$ و همچنین G_{best} راه‌حل ارائه شده توسط یک الگوریتم پس از اجرای برنامه مربوطه باشد. علاوه بر این، مقدار خطای قابل قبول برای این الگوریتم را ϵ در نظر بگیرید. اگر $|f(G_{best}) - f(X^*)| < \epsilon$ ، آن‌گاه این اجرا را موفق ارزیابی می‌کنیم.

جدول ۱. پارامترهای شش استراتژی LF معروف و چهار LF از کلاس استراتژی‌های ELF.

منبع	پارامترهای IW	استراتژی IW	پارامترهای LF	استراتژی LF
[۱۷, ۲]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	$c_1 = c_2 = 2$	CLF
[۲۲, ۱۶]	$\omega = 0.7298$	CIW	$c_1 = c_2 = 1.4962$	CFLF
[۱۹]	---	RIW	---	RLF
[۲۴, ۱۷]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	$c_{1i} = c_{2f} = 2.5$ $c_{1f} = c_{2i} = 0.5$	TVAC
[۲۵, ۱۷]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	---	SAAF
[۱۸, ۱۷]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	$c_{1i} = c_{2f} = 2.5$ $c_{1f} = c_{2i} = 0.5$	ALF
[۲۱]	$\omega_{\max} = 0.95$ $\omega_{\min} = 0.4$ $d_1 = 0.2$ $d_2 = 7$	EDIW	$c_{1i} = c_{2f} = 2.7$ $c_{1f} = c_{2i} = 0.7$	TELF
[۱۷]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	$c_{1i} = 2.8$ $c_{1f} = 0.8$ $c_{2i} = 0.2$ $c_{2f} = 2.2$	RELF
[۱۷]	$\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$	LDIW	$c_{1i} = 2.75$ $c_{1f} = 0.75$ $c_{2i} = 0.25$ $c_{2f} = 2.25$	SELF
[۲۱]	$\omega_{\max} = 0.95$ $\omega_{\min} = 0.4$ $d_1 = 0.2$ $d_2 = 7$	EDIW	$c_{1i} = 2.7$ $c_{1f} = 0.7$ $c_{2i} = 0.7$ $c_{2f} = 2.7$	LELF

تعریف. یک الگوریتم را خاتمه یافته فرض می‌کنیم هرگاه یکی از شرایط زیر محقق شود:

شرط اول توقف: رسیدن به حداکثر تعداد تکرار مجاز $I_{\max}$.

شرط دوم توقف: رسیدن به حداکثر تعداد تکرار مجاز $I_{\max}$ یا رسیدن به خطای قابل قبول ϵ ؛ هر کدام که زودتر اتفاق بیفتند.

تعریف. نسبت تعداد اجراهای موفق (S_{run}) به تعداد کل اجراهای الگوریتم (T_{run})، نرخ موفقیت^۱ (SR) آن الگوریتم نامیده

می‌شود؛ به عبارت دیگر $SR = \frac{S_{run}}{T_{run}} \times 100$. همچنین برای یک الگوریتم مفروض، میانگین خطا^۲ (AE) در یافتن راه‌حل بهینه

تابع معیار $f(\cdot)$ به صورت $AE = \frac{\sum_{run} |f(G_{best}) - f(X^*)|}{T_{run}}$ تعریف می‌شود.

^۱ Success Rate
^۲ Average Error

برای ارزیابی هر الگوریتم در حل هر مسئله، مقادیر نرخ موفقیت (SR)، میانگین تعداد تکرارهای الگوریتم در اجراهای موفق^۱ (ANS)، کمینه تعداد تکرارهای الگوریتم در اجراهای موفق^۲ (MNS)، میانگین خطا (AE)، کمینه خطا^۳ (ME) و انحراف معیار^۴ (STD) خطا در ۱۰۰ مرتبه اجرای الگوریتم به‌عنوان PEC محاسبه و ثبت می‌گردد.

۳-۵- توابع معیار غیرخطی استاندارد

از بین توابع معیار استاندارد، ۲۶ مسئله غیرخطی رایج و پرکاربرد که در [۱۳] به کار رفته، برای ارزیابی کارایی و دقت الگوریتم‌های کلاس ELF-PSO استفاده شده است. با کمک این توابع، می‌توان عملکرد الگوریتم‌های پیشنهادی را با معروف‌ترین نسخه‌های الگوریتم PSO مقایسه کرده و بهترین الگوریتم را انتخاب نمود. این توابع معیار، پیوسته بوده و دارای درجه پیچیدگی‌های متفاوت از تک‌وجهی^۵ تا چندوجهی^۶ هستند.

۴-۵- آزمون‌های آماری ناپارامتری

برای مقایسه نتایج چند الگوریتم و تعیین بهترین الگوریتم از بین آن‌ها، نیاز به ابزاری قوی و قابل اطمینان است که در این مقاله از روش آزمون‌های آماری ناپارامتری^۷ بهره می‌بریم. این روش‌ها دستورالعمل‌هایی ریاضی برای آزمون فرضیه هستند و بارزترین ویژگی آن‌ها این است که هیچ فرضی درباره توزیع احتمال متغیرهای مورد آزمون نیاز ندارند. آزمون ویلکاکسون^۸، آزمون فریدمن^۹ و آزمون بونفرونی-دان^{۱۰} روش‌های ناپارامتری هستند که مورد استفاده قرار خواهند گرفت [۱۳، ۲۷].

۵-۴-۱- آزمون ویلکاکسون

ویلکاکسون یک آزمون فرضیه آماری ناپارامتری است که هرگاه نتوان فرض کرد نتایج دارای توزیع نرمال هستند، می‌تواند به‌عنوان جایگزینی برای آزمون t زوجی^{۱۱} استفاده شود. آزمون ویلکاکسون، خروجی الگوریتم‌ها را به‌صورت جفتی با هم مقایسه می‌کند و نتایج آن را به فرم R^+ و R^- نمایش می‌دهد که به ترتیب نشان‌دهنده جمع رتبه‌های مثبت و منفی یک الگوریتم در مقایسه با الگوریتم دیگر است. ضمناً در این آزمون میزان قابلیت اطمینان هر نتیجه با یک p -value تعیین می‌شود [۱۳].

۵-۴-۲- آزمون فریدمن

فریدمن یک آزمون فرضیه آماری ناپارامتری است و هدفش این است که تعیین کند آیا بین الگوریتم‌هایی که بر روی یک پایگاه داده مشخص اعمال شده‌اند، اختلاف معناداری وجود دارد یا خیر. این آزمون به الگوریتم‌های شرکت‌کننده در رقابت، یک رتبه اختصاص می‌دهد. در مسائل کمینه‌سازی، بهترین الگوریتم حائز رتبه کمینه می‌شود [۱۳].

¹ Average Number of Iterations of Successful Runs

² Minimum Number of Iterations of Successful Runs

³ Minimum Error

⁴ Standard Deviation

⁵ Unimodal

⁶ Multimodal

⁷ Non-Parametric Statistical Tests

⁸ Wilcoxon Sign Rank Test

⁹ Friedman Test

¹⁰ Bonferroni-Dunn Test

¹¹ Paired T-Test

۵-۳-۴-۵ آزمون بونفرونی-دان

آزمون بونفرونی-دان برای مقایسه یک الگوریتم با سایر الگوریتم‌ها به کار می‌رود. بر اساس آزمون بونفرونی-دان، عملکرد دو الگوریتم به طور معناداری متفاوت است. اگر تفاوت میانگین رتبه‌های آن‌ها حداقل به اندازه اختلاف بحرانی^۱ (CD) زیر باشد:

$$CD_{\alpha} = q_{\alpha} \sqrt{\frac{N_i(N_i + 1)}{6N_f}} \quad (14)$$

که N_i و N_f به ترتیب تعداد الگوریتم‌های شرکت کننده در رقابت و تعداد توابع معیار هستند. همچنین مقدار بحرانی^۲ q_{α} در سطح احتمال α ، به صورت زیر داده شده است [۱۳]:

$$q_{0.05} = 2.773, \quad q_{0.1} = 2.539 \quad (15)$$

معمولاً نتیجه این آزمون که نمایان گر تفاوت بین میانگین رتبه‌های الگوریتم‌ها است را با نمودار میله‌ای^۳ نمایش می‌دهند.

۶- ارزیابی عملکرد الگوریتم‌های ELF-PSO

این بخش به ارزیابی گونه‌های مختلف الگوریتم ELF-PSO در مقابل الگوریتم‌های PSO مبتنی بر معروف‌ترین و پرکاربردترین ضرائب یادگیری می‌پردازد. به همین منظور، باید ثابت کنیم که الگوریتم‌های ELF-PSO پیشنهادی، توانایی رقابت با سایر الگوریتم‌های بهینه‌سازی مشابه را دارا هستند. بنابراین پس از اعمال این الگوریتم‌ها بر روی توابع معیار، نتایج عددی آن‌ها را نمایش می‌دهیم. سپس با استفاده از آزمون‌های ویلکاکسون، فریدمن و بونفرونی-دان این نتایج عددی را مورد تجزیه و تحلیل آماری قرار می‌دهیم. در ادامه این بخش، نمودار میله‌ای بونفرونی-دان و گراف همگرایی این الگوریتم‌ها را ارائه و تشریح خواهیم کرد تا بهترین الگوریتم‌های موجود در این رقابت معین شوند.

۶-۱- نتایج عددی اجرای الگوریتم‌ها

بر اساس توابع معیار استاندارد [۱۳]، یک مطالعه مقایسه‌ای جامع با استفاده از شش PEC بین ۱۰ گونه الگوریتم PSO صورت گرفت. نتایج محاسباتی در جدول ۲ تا جدول ۷ آمده است. لازم به توضیح است که در جدول ۲ و جدول ۳ منظور از v عددی بزرگ‌تر از $I_{\max}$ است. ضمناً خطای قابل قبول برای توابع f_{15} و f_{20} مقدار $\varepsilon = 5$ ، برای توابع $f_2, f_3, f_4, f_{10}, f_{19}, f_{21}, f_{24}$ و f_{25} مقدار $\varepsilon = 10^{-1}$ ، برای توابع f_5, f_{16} و f_{23} مقدار $\varepsilon = 10^{-10}$ و برای سایر توابع معیار $\varepsilon = 10^{-30}$ فرض شده است.

با توجه به خروجی الگوریتم‌ها در جدول ۲ تا جدول ۷، برتری گونه‌های مختلف الگوریتم ELF-PSO از نظر سرعت و دقت نسبت به سایر الگوریتم‌ها مشهود است؛ اما در بخش بعد با استفاده از تحلیل‌های آماری این مشاهدات را ثابت خواهیم کرد و نشان می‌دهیم که الگوریتم‌های پیشنهادی نتایج را بهبود می‌دهند.

¹ Critical Difference

² Critical Value

³ Bar Charts

جدول ۲. مقایسه نرخ موفقیت، میانگین و کمینه تعداد تکرارهای الگوریتم در اجراهای موفق.

مفروضات: توقف الگوریتم با شرط دوم صورت گرفته است و $D = 10$, $I_{\max} = 1000$

f_{13}	f_{12}	f_{11}	f_{10}	f_9	f_8	f_7	f_6	f_5	f_4	f_3	f_2	f_1	PEC	LF
42	0	0	5	0	19	42	0	100	0	2	77	55	SR	CLF
727	ν	ν	517	ν	993	987	ν	882	ν	677	696	988	ANS	
615	ν	ν	434	ν	987	962	ν	849	ν	512	514	959	MNS	
56	12	0	7	100	100	96	100	100	0	9	83	100	SR	CFLF
448	963	ν	78	943	578	561	609	403	ν	340	180	557	ANS	
318	886	ν	59	877	533	524	568	377	ν	66	73	527	MNS	
23	22	100	10	100	100	95	100	93	0	32	64	100	SR	RLF
528	291	357	39	676	204	203	213	146	ν	298	75	193	ANS	
99	254	313	22	612	186	175	189	122	ν	19	22	166	MNS	
60	79	100	1	100	100	100	100	100	2	18	85	100	SR	TVAC
633	827	915	270	801	630	618	646	523	428	726	386	616	ANS	
469	762	884	270	769	608	597	625	503	379	248	237	591	MNS	
53	7	3	3	100	100	99	100	99	0	15	70	100	SR	SAAF
443	622	757	98	544	404	396	419	327	ν	503	148	395	ANS	
382	568	642	88	496	391	384	408	309	ν	121	97	382	MNS	
96	16	0	4	95	100	96	100	100	0	87	64	100	SR	ALF
748	968	ν	331	969	862	851	880	704	ν	911	399	845	ANS	
572	903	ν	311	939	827	818	851	676	ν	417	305	802	MNS	
80	72	100	0	100	100	100	100	100	2	61	99	100	SR	TELF
525	417	453	ν	418	281	276	290	225	296	756	154	274	ANS	
223	387	433	ν	397	275	270	283	218	209	117	101	268	MNS	
79	79	94	4	100	100	100	100	100	7	75	95	100	SR	RELF
645	674	728	247	620	503	497	516	424	434	774	305	494	ANS	
442	539	694	222	600	495	485	505	413	294	311	192	480	MNS	
98	64	87	4	100	100	100	100	100	0	92	86	100	SR	SELF
611	685	761	202	611	501	494	516	416	ν	687	283	491	ANS	
398	654	717	192	588	489	484	505	403	ν	254	190	480	MNS	
77	79	100	1	100	100	100	100	100	3	61	96	100	SR	LELF
561	447	496	879	459	304	299	314	241	288	801	176	296	ANS	
222	415	481	879	429	296	289	305	233	241	207	106	284	MNS	

جدول ۳. مقایسه نرخ موفقیت، میانگین و کمینه تعداد تکرارهای الگوریتم در اجراهای موفق.

مفروضات: توقف الگوریتم با شرط دوم صورت گرفته است و $D = 10$ ، $I_{\max} = 1000$

f_{26}	f_{25}	f_{24}	f_{23}	f_{22}	f_{21}	f_{20}	f_{19}	f_{18}	f_{17}	f_{16}	f_{15}	f_{14}	PEC	LF
100	66	96	81	18	4	98	99	16	40	92	55	0	SR	CLF
738	669	627	754	992	409	235	497	991	992	877	671	ν	ANS	
704	424	339	716	979	307	38	444	978	975	766	260	ν	MNS	
100	70	81	70	100	1	67	95	100	100	86	21	100	SR	CFLF
249	195	205	258	613	62	67	102	587	577	356	283	716	ANS	
221	69	88	230	562	62	18	84	548	543	264	81	671	MNS	
100	44	76	61	76	5	18	75	96	86	28	4	100	SR	RLF
92	158	181	110	203	25	173	45	216	204	299	21	249	ANS	
75	50	36	75	171	16	10	28	184	180	125	17	213	MNS	
100	82	93	99	91	10	97	100	100	100	94	46	100	SR	TVAC
444	349	389	421	666	291	192	262	634	623	504	385	703	ANS	
406	233	243	399	639	171	59	227	610	604	438	209	684	MNS	
100	44	79	42	39	0	76	85	100	99	25	2	100	SR	SAAF
231	164	203	243	449	ν	99	139	409	399	325	192	470	ANS	
208	107	135	226	420	ν	35	113	397	384	249	115	450	MNS	
100	50	98	76	88	0	97	100	99	100	42	75	100	SR	ALF
588	375	361	592	829	ν	187	322	862	852	597	405	939	ANS	
534	267	225	564	802	ν	48	285	816	812	546	247	904	MNS	
100	52	85	97	74	16	78	100	100	100	70	63	100	SR	TELF
203	170	220	182	336	173	99	111	287	280	247	261	325	ANS	
170	113	81	167	294	87	39	102	278	273	193	99	316	MNS	
100	74	89	99	66	17	82	100	100	100	73	64	100	SR	RELF
460	307	347	350	566	376	201	230	509	500	439	437	560	ANS	
342	237	192	336	519	198	83	203	492	488	367	243	550	MNS	
100	62	92	96	66	0	71	100	100	100	72	25	100	SR	SELF
335	266	336	333	550	ν	243	209	507	498	402	309	565	ANS	
312	199	242	315	516	ν	93	187	496	488	349	101	551	MNS	
100	61	83	99	90	10	80	100	100	100	77	54	100	SR	LELF
201	199	192	188	349	112	96	114	310	302	260	257	353	ANS	
181	114	99	174	316	77	30	102	301	293	193	110	343	MNS	

جدول ۴. مقایسه میانگین خطا، کمینه خطا و انحراف معیار خطا.

مفروضات: توقف الگوریتم با شرط اول صورت گرفته است و $D = 10$ ، $I_{\max} = 1000$

f_7	f_6	f_5	f_4	f_3	f_2	f_1	PEC	LF
4.22e-30	3.57e-27	7.64e-15	6.62e+00	3.23e+00	9.13e-02	7.70e-30	AE	CLF
1.50e-32	1.86e-30	4.44e-15	9.95e-01	2.98e-02	1.97e-02	7.28e-34	ME	
1.12e-29	6.27e-27	3.03e-15	3.61e+00	1.36e+00	5.16e-02	2.68e-29	STD	
1.50e-32	6.58e-50	4.44e-15	9.55e+00	1.68e+00	6.41e-02	1.03e-52	AE	CFLF
1.50e-32	2.80e-53	4.44e-15	1.99e+00	6.88e-03	2.21e-02	5.12e-56	ME	
5.62e-48	1.75e-49	1.00e-300	3.88e+00	1.88e+00	3.23e-02	2.36e-52	STD	
1.50e-32	2.07e-147	5.15e-15	1.04e+01	1.88e+00	1.02e-01	2.52e-151	AE	RLF
1.50e-32	1.36e-157	4.44e-15	2.98e+00	6.25e-04	9.86e-03	1.91e-161	ME	
5.62e-48	5.78e-147	1.46e-15	5.12e+00	2.10e+00	5.67e-02	9.40e-151	STD	
1.50e-32	4.28e-64	5.51e-15	3.68e+00	6.03e-01	6.74e-02	3.23e-67	AE	TVAC
1.50e-32	1.11e-70	4.44e-15	9.95e-01	1.17e-02	7.40e-03	1.05e-70	ME	
5.62e-48	1.54e-63	1.67e-15	2.12e+00	1.19e+00	3.09e-02	9.44e-67	STD	
1.50e-32	9.60e-106	6.75e-15	1.13e+01	1.46e+00	1.03e-01	3.94e-86	AE	SAAF
1.50e-32	6.49e-125	4.44e-15	2.98e+00	1.42e-03	4.43e-02	3.82e-129	ME	
5.62e-48	3.62e-105	1.74e-15	5.24e+00	1.78e+00	4.89e-02	1.76e-85	STD	
1.75e-32	1.87e-40	4.97e-15	7.61e+00	1.05e+00	1.13e-01	1.48e-43	AE	ALF
1.50e-32	2.21e-43	4.44e-15	2.98e+00	2.11e-02	3.44e-02	8.40e-47	ME	
1.10e-32	5.52e-40	1.30e-15	3.10e+00	1.77e+00	4.30e-02	3.56e-43	STD	
1.50e-32	1.03e-137	5.51e-15	2.59e+00	2.21e-01	3.71e-02	1.33e-141	AE	TELF
1.50e-32	1.62e-142	4.44e-15	1.00e-300	3.96e-04	9.86e-03	1.84e-146	ME	
1.11e-47	5.38e-137	1.66e-15	1.62e+00	4.10e-01	1.74e-02	4.83e-141	STD	
1.50e-32	2.15e-96	5.74e-15	2.29e+00	9.84e-02	4.44e-02	3.06e-99	AE	RELf
1.50e-32	1.07e-99	4.44e-15	1.00e-300	4.98e-06	1.00e-300	8.35e-103	ME	
1.11e-47	5.14e-96	1.74e-15	1.26e+00	1.47e-01	2.78e-02	1.10e-98	STD	
1.50e-32	1.67e-83	5.51e-15	3.22e+00	4.08e-01	5.50e-02	6.78e-86	AE	SELF
1.50e-32	8.72e-88	4.44e-15	1.00e-300	3.56e-06	7.40e-03	6.42e-90	ME	
1.11e-47	5.06e-83	1.66e-15	1.54e+00	1.22e+00	1.86e-02	2.88e-85	STD	
1.50e-32	9.09e-115	4.44e-15	3.78e+00	1.09e-01	5.41e-02	7.55e-117	AE	LELF
1.50e-32	4.48e-121	4.44e-15	1.99e+00	4.96e-03	1.00e-300	1.08e-123	ME	
1.00e-300	2.55e-114	1.00e-300	1.68e+00	8.23e-02	2.91e-02	2.09e-116	STD	

جدول ۵. مقایسه میانگین خطا، کمینه خطا و انحراف معیار خطا.

مفروضات: توقف الگوریتم با شرط اول صورت گرفته است و $D = 10$, $I_{\max} = 1000$

f_{14}	f_{13}	f_{12}	f_{11}	f_{10}	f_9	f_8	PEC	LF
3.15e-21	3.19e-10	7.29e-16	1.60e-16	6.33e-01	3.98e-14	9.94e-29	AE	CLF
2.21e-25	1.00e-300	3.45e-18	7.69e-18	6.27e-23	4.74e-16	1.70e-31	ME	
1.03e-20	1.43e-09	9.44e-16	2.51e-16	1.49e-01	4.48e-14	1.31e-28	STD	
8.54e-46	6.51e-20	1.30e-15	4.63e-28	6.67e-01	3.83e-32	3.35e-52	AE	CFLF
1.40e-47	1.00e-300	1.03e-36	1.65e-29	6.67e-01	6.11e-37	2.83e-54	ME	
1.07e-45	1.02e-19	2.34e-15	6.88e-28	1.22e-10	6.27e-32	7.92e-52	STD	
9.30e-138	5.43e-07	8.63e-16	1.04e-83	4.67e-01	4.32e-42	6.73e-145	AE	RLF
4.20e-150	1.00e-300	5.69e-200	6.00e-89	4.56e-31	1.30e-49	9.68e-159	ME	
4.16e-137	1.72e-06	9.54e-16	3.55e-83	3.13e-01	1.56e-41	2.53e-144	STD	
8.27e-59	1.63e-19	1.86e-16	9.01e-35	6.00e-01	2.37e-43	3.01e-66	AE	TVAC
5.09e-64	1.00e-300	4.43e-57	1.20e-36	1.97e-31	4.51e-48	6.11e-71	ME	
3.45e-58	4.33e-19	2.89e-16	1.04e-34	2.05e-01	6.50e-43	8.74e-66	STD	
9.50e-85	3.20e-09	5.28e-10	3.54e-16	6.67e-01	6.69e-45	1.86e-84	AE	SAAF
9.95e-121	1.00e-300	2.22e-16	2.06e-29	6.67e-01	3.91e-65	3.12e-125	ME	
4.25e-84	1.43e-08	2.01e-09	1.51e-15	2.04e-16	2.99e-44	8.33e-84	STD	
9.24e-36	2.17e-20	6.27e-16	1.38e-16	6.33e-01	3.25e-31	4.74e-42	AE	ALF
4.05e-39	1.00e-300	5.85e-36	1.24e-28	1.97e-31	5.57e-34	1.01e-46	ME	
1.57e-35	6.67e-20	6.69e-16	6.15e-16	1.49e-01	1.00e-30	1.79e-41	STD	
8.02e-131	5.06e-20	1.28e-16	1.05e-32	6.22e-01	1.86e-83	8.94e-141	AE	TELF
3.48e-137	1.00e-300	2.48e-107	3.94e-73	1.97e-31	3.72e-91	2.81e-145	ME	
2.98e-130	9.33e-20	2.67e-16	5.78e-32	1.69e-01	7.25e-83	2.38e-140	STD	
2.33e-89	7.23e-20	7.07e-15	5.05e-23	6.67e-01	3.52e-66	7.42e-98	AE	RELf
1.36e-94	1.00e-300	2.88e-71	7.52e-51	6.67e-01	4.52e-71	5.80e-103	ME	
1.12e-88	1.74e-19	3.85e-14	2.76e-22	4.61e-17	1.33e-65	1.81e-97	STD	
1.27e-77	1.00e-300	2.83e-16	1.24e-24	6.22e-01	7.25e-61	2.39e-84	AE	SELF
4.32e-81	1.00e-300	1.07e-66	1.65e-46	3.20e-31	2.02e-67	9.65e-89	ME	
2.15e-77	1.00e-300	5.86e-16	6.79e-24	1.69e-01	2.23e-60	7.20e-84	STD	
1.07e-107	1.00e-300	5.11e-16	7.14e-60	6.67e-01	1.13e-69	5.23e-115	AE	LELF
1.85e-114	1.00e-300	4.83e-93	7.21e-62	6.67e-01	4.25e-75	1.13e-121	ME	
3.29e-107	1.00e-300	1.61e-15	1.21e-59	1.17e-16	3.55e-69	8.51e-115	STD	

جدول ۶. مقایسه میانگین خطا، کمینه خطا و انحراف معیار خطا.

مفروضات: توقف الگوریتم با شرط اول صورت گرفته است و $D = 10$ ، $I_{\max} = 1000$

f_{21}	f_{20}	f_{19}	f_{18}	f_{17}	f_{16}	f_{15}	PEC	LF
6.09e-01	2.52e+00	7.50e-02	4.49e-28	4.59e-29	3.94e-13	4.30e+00	AE	CLF
1.46e-01	1.11e+00	1.00e-300	2.83e-32	1.13e-31	1.00e-300	1.00e-300	ME	
3.80e-01	1.03e+00	3.35e-01	1.63e-27	1.42e-28	1.70e-12	1.72e+00	STD	
7.28e-01	3.69e+00	7.92e-02	1.35e-32	4.71e-32	3.59e-01	5.75e+00	AE	CFLF
1.92e-01	6.71e-01	1.00e-300	1.35e-32	4.71e-32	1.00e-300	2.00e+00	ME	
5.16e-01	1.46e+00	3.52e-01	2.81e-48	5.62e-48	8.44e-01	1.55e+00	STD	
6.25e-01	6.16e+00	4.23e-02	2.75e-03	3.11e-02	1.97e+00	8.80e+00	AE	RLF
2.40e-01	3.09e+00	1.00e-300	1.35e-32	4.71e-32	1.00e-300	5.00e+00	ME	
3.78e-01	2.06e+00	8.55e-02	4.88e-03	9.57e-02	3.10e+00	2.63e+00	STD	
3.48e-01	3.12e+00	5.24e-05	1.35e-32	4.71e-32	1.12e-01	4.95e+00	AE	TVAC
1.72e-06	3.73e-01	1.00e-300	1.35e-32	4.71e-32	1.00e-300	3.00e+00	ME	
3.39e-01	1.34e+00	2.02e-04	2.81e-48	5.62e-48	5.02e-01	9.99e-01	STD	
4.62e+00	2.58e+00	7.28e-02	1.35e-32	4.71e-32	3.01e+00	7.20e+00	AE	SAAF
3.57e+00	7.00e-01	1.00e-300	1.35e-32	4.71e-32	1.00e-300	2.00e+00	ME	
4.38e-01	1.17e+00	2.27e-01	2.81e-48	5.62e-48	4.37e+00	2.71e+00	STD	
6.19e+00	2.70e+00	7.57e-04	1.35e-32	4.79e-32	9.87e-01	3.65e+00	AE	ALF
4.98e+00	5.91e-01	1.00e-300	1.35e-32	4.71e-32	1.00e-300	1.00e-300	ME	
4.59e-01	1.25e+00	3.23e-03	2.81e-48	3.46e-33	1.32e+00	1.76e+00	STD	
3.56e-01	3.71e+00	6.87e-04	1.35e-32	4.71e-32	1.65e-01	4.60e+00	AE	TELF
1.72e-06	1.51e+00	1.00e-300	1.35e-32	4.71e-32	1.00e-300	3.00e+00	ME	
1.89e-01	1.18e+00	3.00e-03	5.57e-48	1.67e-47	5.20e-01	1.33e+00	STD	
3.01e-01	3.10e+00	3.17e-04	1.35e-32	4.71e-32	4.94e-01	4.30e+00	AE	RELf
1.72e-06	1.38e+00	1.00e-300	1.35e-32	4.71e-32	1.00e-300	3.00e+00	ME	
2.64e-01	1.19e+00	1.35e-03	5.57e-48	1.67e-47	1.08e+00	8.37e-01	STD	
6.04e+00	4.59e+00	3.22e-04	1.35e-32	4.71e-32	7.63e-01	4.97e+00	AE	SELF
5.39e+00	6.49e-01	1.00e-300	1.35e-32	4.71e-32	1.00e-300	3.00e+00	ME	
3.38e-01	1.46e+00	1.06e-03	5.57e-48	1.67e-47	1.58e+00	8.09e-01	STD	
4.61e-01	3.90e+00	6.94e-05	1.35e-32	4.71e-32	2.24e-01	4.00e+00	AE	LELF
6.69e-02	1.83e+00	1.00e-300	1.35e-32	4.71e-32	1.00e-300	3.00e+00	ME	
4.72e-01	1.73e+00	1.52e-04	2.88e-48	1.15e-47	5.70e-01	1.05e+00	STD	

جدول ۷. مقایسه میانگین خطا، کمینه خطا و انحراف معیار خطا.

مفروضات: توقف الگوریتم با شرط اول صورت گرفته است و $D = 10$ ، $I_{\max} = 1000$.

f_{26}	f_{25}	f_{24}	f_{23}	f_{22}	PEC	LF
1.00e-300	1.20e-01	4.40e-02	2.25e+00	8.86e-15	AE	CLF
1.00e-300	9.99e-02	5.50e-03	6.88e-29	1.00e-300	ME	
1.00e-300	4.10e-02	4.07e-02	1.01e+01	1.60e-14	STD	
1.00e-300	1.50e-01	7.43e-02	2.22e+01	1.00e-300	AE	CFLF
1.00e-300	9.99e-02	2.60e-03	3.98e-52	1.00e-300	ME	
1.00e-300	5.13e-02	1.13e-01	3.33e+01	1.00e-300	STD	
1.00e-300	1.65e-01	7.30e-02	5.63e+01	9.77e-16	AE	RLF
1.00e-300	9.99e-02	7.36e-03	9.47e-152	1.00e-300	ME	
1.00e-300	5.87e-02	1.03e-01	7.53e+01	3.32e-15	STD	
1.00e-300	1.15e-01	3.65e-02	9.63e-64	1.00e-300	AE	TVAC
1.00e-300	9.99e-02	1.65e-03	1.04e-67	1.00e-300	ME	
1.00e-300	3.66e-02	8.00e-02	2.20e-63	1.00e-300	STD	
1.00e-300	1.50e-01	5.87e-02	3.30e+01	3.89e-12	AE	SAAF
1.00e-300	9.99e-02	6.08e-03	1.13e-118	1.00e-300	ME	
1.00e-300	5.13e-02	5.63e-02	5.29e+01	1.14e-11	STD	
1.00e-300	1.65e-01	3.57e-02	7.70e+00	1.33e-16	AE	ALF
1.00e-300	9.99e-02	1.76e-03	6.86e-44	1.00e-300	ME	
1.00e-300	4.89e-02	4.02e-02	1.91e+01	2.92e-16	STD	
1.00e-300	1.43e-01	3.47e-02	1.98e+00	2.66e-16	AE	TELF
1.00e-300	9.99e-02	2.14e-03	2.94e-143	1.00e-300	ME	
1.00e-300	5.04e-02	5.26e-02	1.09e+01	7.43e-16	STD	
1.00e-300	1.27e-01	3.94e-02	4.24e-95	5.29e-13	AE	RELF
1.00e-300	9.99e-02	3.44e-03	8.69e-99	1.00e-300	ME	
1.00e-300	4.50e-02	3.62e-02	1.39e-94	2.38e-12	STD	
1.00e-300	1.43e-01	2.97e-02	3.51e+00	2.96e-16	AE	SELF
1.00e-300	9.99e-02	3.42e-03	3.04e-86	1.00e-300	ME	
1.00e-300	5.04e-02	2.54e-02	1.35e+01	8.36e-16	STD	
1.00e-300	1.20e-01	5.08e-02	2.90e-111	8.88e-17	AE	LELF
1.00e-300	9.99e-02	5.97e-03	8.12e-118	1.00e-300	ME	
1.00e-300	4.22e-02	6.03e-02	7.89e-111	1.87e-16	STD	

۶-۲- تحلیل آماری نتایج عددی

در این بخش با استفاده از آزمون‌های آماری ناپارامتری و ترسیم نمودارها به تحلیل نتایج عددی می‌پردازیم و بهترین الگوریتم‌ها را معرفی می‌نمائیم. در انجام این آزمون‌ها برای میانگین خطا، کمینه خطا و انحراف معیار خطا از مقیاس لگاریتمی استفاده می‌کنیم.

۶-۲-۱- نتایج آزمون ویلکاکسون

در این آزمون، طبق فرض H_0 الگوریتم پیشنهادی ELF-PSO باعث بهبود راه‌حل بهینه نشده است. ضمناً H_1 به عنوان فرض مقابل و سطح معنادر بودن آزمون $\alpha = 0.05$ تعریف می‌شود. بنابراین اگر $p\text{-value} < 0.05$ آن‌گاه فرض H_0 رد و فرض H_1 پذیرفته می‌شود؛ یعنی به لحاظ آماری تأیید می‌شود که الگوریتم ELF-PSO عملکرد بهتری نسبت به الگوریتم رقیب دارد. نتیجه اعمال آزمون ویلکاکسون بر روی جداول ۲ تا ۷ در جداول ۸ تا ۱۱ ارائه شده است. در جداول ۸ و ۹، به ترتیب نتایج اعمال آزمون ویلکاکسون بر روی داده‌های نرخ موفقیت و کمینه تکرارها در اجراهای موفق، ارائه شده‌اند. با توجه به این نتایج و بر اساس معیار نرخ موفقیت، استراتژی‌های TELF و SELF به طور معناداری بهتر از CLF، RLF و SAAF هستند. همچنین استراتژی‌های RELF و LELF با $p\text{-value} < 0.05$ ، بر استراتژی‌های CLF، CFLF، RLF و SAAF فائق می‌آیند. از نظر کمینه تعداد تکرارها در اجراهای موفق، استراتژی‌های TELF و LELF به طور معناداری بهتر از CLF، CFLF، TVAC، SAAF و ALF هستند. علاوه بر این، استراتژی‌های RELF و SELF با $p\text{-value} < 0.05$ ، بر استراتژی‌های CLF، TVAC، ALF و برتری دارند. در جداول ۱۰ و ۱۱، نتایج آزمون ویلکاکسون بر روی میانگین و کمینه خطا نشان داده شده است. از نظر معیار میانگین خطا، این جداول بیانگر برتری استراتژی‌های TELF، RELF و LELF بر CLF، CFLF، SAAF و ALF هستند. همچنین استراتژی SELF با $p\text{-value} < 0.05$ ، بر استراتژی‌های CLF، SAAF، ALF پیروز می‌شود. از نظر کمینه خطا، آزمون ویلکاکسون رأی به برتری استراتژی‌های TELF، RELF، SELF و LELF نسبت به CLF، CFLF، TVAC، SAAF و ALF می‌دهد.

۶-۲-۲- نتایج آزمون فریدمن

در این قسمت، فرض H_0 آزمون فریدمن این است که رتبه الگوریتم‌ها از جمله رتبه الگوریتم پیشنهادی $ELF - PSO$ ، تفاوتی با همدیگر ندارند و فرض H_1 به عنوان فرض مقابل است. همچنین سطح معنادر بودن آزمون $\alpha = 0.05$ تعریف می‌شود. نتیجه اعمال آزمون فریدمن بر روی جداول ۲ تا ۷ در جداول ۱۲ و ۱۳ آورده شده است. در همه این جداول $p\text{-value}$ کمتر از سطح معناداری $\alpha = 0.01$ است، بنابراین در خروجی همه آزمون‌های فریدمن، تفاوت معناداری بین رتبه‌های الگوریتم‌ها مشاهده می‌شود. با استفاده از جدول ۱۲، استراتژی RLF از نظر میانگین و کمینه تعداد تکرارها در اجراهای موفق، بهترین عملکرد را دارد اما از لحاظ نرخ موفقیت، رتبه ضعیف ۸ را داراست؛ به عبارت دیگر، با وجود اینکه الگوریتم RLF-PSO دارای سرعت خوبی است، ولی شانس کمی برای یافتن جواب بهینه سراسری دارد. از طرف دیگر استراتژی RELF از نظر نرخ موفقیت، دارای بهترین رتبه است، اما در مقایسه با روش‌های دیگر، سرعت همگرایی خیلی بالایی ندارد. البته چنین پدیده‌ای قابل درک و پذیرش است، زیرا طبق قضیه «هیچ ناهاری مجانی نیست» یک الگوریتم بهینه‌سازی نمی‌تواند نسبت به بقیه الگوریتم‌ها از هر نظر یا روی هر نوع مسأله‌ای، عملکرد بهتری داشته باشد. بر اساس جدول ۱۲ می‌توان نتیجه گرفت که استراتژی TELF بهتر از بقیه رقبا است، زیرا از لحاظ هر دو معیار میانگین و کمینه تعداد تکرارها در اجراهای موفق، دارای رتبه ۲ بوده و رتبه ۴ را از لحاظ نرخ موفقیت داراست.

در واقع بر اساس آزمون فریدمن، الگوریتم TELF-PSO سرعت همگرایی بالایی داشته و از نرخ موفقیت قابل قبولی برخوردار است. طبق جدول ۱۳ استراتژی TELF از نظر میانگین و کمینه خطا، بهترین عملکرد را در بین همه استراتژی‌ها داراست. بنابراین می‌توان نتیجه گرفت آزمون فریدمن ثابت می‌کند که الگوریتم TELF-PSO دقیق‌ترین جواب را در بین مابقی رقبا ارائه می‌دهد. با توجه به تحلیل‌های صورت گرفته بر روی نتایج آزمون فریدمن، در مجموع می‌توان نشان داد که الگوریتم‌های کلاس ELF-PSO از نظر دقت، سرعت و شانس موفقیت، به‌طور معناداری بهتر از الگوریتم‌های CLF-PSO، CFLF-PSO، RLF-PSO، TVAC-PSO، SAAF-PSO و ALF-PSO عمل می‌کنند.

۶-۲-۳- نتایج آزمون بونفرونی-دان

تفاوت در بین رتبه‌بندی میانگین گونه‌های مختلف الگوریتم PSO، توسط نمودارهای میله‌ای بونفرونی-دان در شکل ۲ نمایش داده شده است. در این شکل، نمودارهای میله‌ای بونفرونی-دان بر اساس معیارهای میانگین و کمینه تعداد تکرارها در اجراهای موفق، نشان می‌دهند که استراتژی RLF بهترین سرعت را در همگرایی به بهینه سراسری دارد. البته این شکل ثابت می‌کند که الگوریتم RLF-PSO نرخ موفقیت مناسبی ندارد. همچنین می‌توان مشاهده کرد که استراتژی‌های CLF، CFLF، TVAC، SAAF، ALF و SELF در مقایسه با سایر روش‌ها، بدترین سرعت همگرایی را دارند. از دیدگاه نرخ موفقیت، بهترین عملکرد متعلق به استراتژی RLF و بدترین نتیجه متعلق به SAAF است. از طرف دیگر، این نمودارها نشان می‌دهند که استراتژی RELF از لحاظ میانگین و کمینه تعداد تکرارها در اجراهای موفق، عملکرد خیلی خوبی ندارد. بنابراین استراتژی‌های RLF و RELF به دلیل نقاط ضعفی که بیان کردیم، بهترین گزینه‌ها نیستند. با این وجود استراتژی TELF عملکرد مناسبی از نظر سرعت همگرایی و نرخ موفقیت دارد. از دیدگاه میانگین خطا، نمودارهای میله‌ای بونفرونی-دان نشان می‌دهند که استراتژی‌های CLF، CFLF، RLF، SAAF و ALF بدترین عملکرد را دارند؛ در حالی که الگوریتم TELF-PSO بهترین جواب‌ها را ارائه می‌دهد. همچنین از نظر کمینه خطا، استراتژی‌های CLF، CFLF، ALF و بدترین گزینه‌ها هستند و TELF بهترین نتیجه را می‌دهد. از نظر انحراف معیار خطا، استراتژی‌های CLF، CFLF، RLF، SAAF و ALF بدترین عملکرد را دارند و LELF بهترین گزینه است. با توجه به نتایج فوق، بر اساس نمودارهای میله‌ای بونفرونی-دان می‌توان دریافت که استراتژی‌های کلاس ELF در بین همه استراتژی‌های رقیب، بهترین عملکرد را دارا هستند.

۶-۳- تحلیل گراف همگرایی

گراف همگرایی استراتژی‌های LELF، TELF، RELF و SELF در شکل ۳ ترسیم شده است. معیار توقف این گراف‌ها شرط دوم است و حداکثر تعداد تکرار مجاز $I_{\max} = 3000$ است. با استفاده از گراف همگرایی، می‌توان نتیجه گرفت که سرعت همگرایی استراتژی‌های کلاس ELF، سریع‌تر از سایر استراتژی‌ها است. از طرف دیگر، بر اساس گراف‌های همگرایی موجود در شکل مشاهده می‌شود جوابی که توسط الگوریتم‌های کلاس ELF-PSO به دست می‌آید، بسیار دقیق‌تر از جواب الگوریتم‌های معروف CLF-PSO، CFLF-PSO، RLF-PSO، TVAC-PSO، SAAF-PSO و ALF-PSO است.

جدول ۸. نتایج آزمون ویلکاکسون بر روی نرخ موفقیت بر اساس جدول ۲ و جدول ۳.

نرخ موفقیت						-----	
ALF	SAAF	TVAC	RLF	CFLF	CLF	نتیجه	الگوریتم
106	184	39.5	180.5	117.5	265.5	R^+	TELF
65	6	51.5	9.5	35.5	34.5	R^-	
0.372	0.000	0.675	0.001	0.052	0.001	p-value	
106.5	190	48	196	131	276	R^+	RELF
46.5	0	43	14	22	24	R^-	
0.155	0.000	0.861	0.001	0.010	0.000	p-value	
79.5	148	35	170.5	102.5	235.5	R^+	SELF
40.5	5	85	19.5	33.5	40.5	R^-	
0.268	0.001	0.155	0.002	0.074	0.003	p-value	
111.5	186	26.5	183	125	265	R^+	LELF
59.5	4	28.5	7	28	35	R^-	
0.257	0.000	0.919	0.000	0.022	0.001	p-value	

جدول ۹. نتایج آزمون ویلکاکسون بر روی کمینه تکرارها در اجراهای موفق بر اساس جدول ۲ و جدول ۳.

کمینه تکرارها در اجراهای موفق						-----	
ALF	SAAF	TVAC	RLF	CFLF	CLF	نتیجه	الگوریتم
327	318	325	49	289.5	338	R^+	TELF
24	33	26	302	61.5	13	R^-	
0.000	0.000	0.000	0.001	0.004	0.000	p-value	
347	52	326.5	27	148.5	349	R^+	RELF
4	299	24.5	324	202.5	2	R^-	
0.000	0.002	0.000	0.000	0.493	0.000	p-value	
297	1	294.5	1	128	299	R^+	SELF
3	299	56.5	324	197	26	R^-	
0.000	0.000	0.003	0.000	0.353	0.000	p-value	
327	307	325	48	284	341	R^+	LELF
24	44	26	303	67	10	R^-	
0.000	0.001	0.000	0.001	0.006	0.000	p-value	

جدول ۱۰. نتایج آزمون ویلکاکسون بر روی میانگین خطا بر اساس جدول ۴ تا جدول ۷.

میانگین خطا						-----	
ALF	SAAF	TVAC	RLF	CFLF	CLF	نتیجه	الگوریتم
257	249	144	202	226	293	R^+	TELF
43	4	87	98	27	32	R^-	
0.002	0.000	0.322	0.137	0.001	0.000	p-value	
234	212	165	166	174	255	R^+	RELF
66	19	88	134	57	45	R^-	
0.016	0.001	0.211	0.648	0.042	0.003	p-value	
263	190	120	178	185	275	R^+	SELF
37	63	111	122	68	50	R^-	
0.001	0.039	0.876	0.424	0.058	0.002	p-value	
278	228	168	199	189	274	R^+	LELF
22	3	85	101	21	26	R^-	
0.000	0.000	0.178	0.162	0.002	0.000	p-value	

جدول ۱۱. نتایج آزمون ویلکاکسون بر روی کمینه خطا بر اساس جدول ۴ تا جدول ۷.

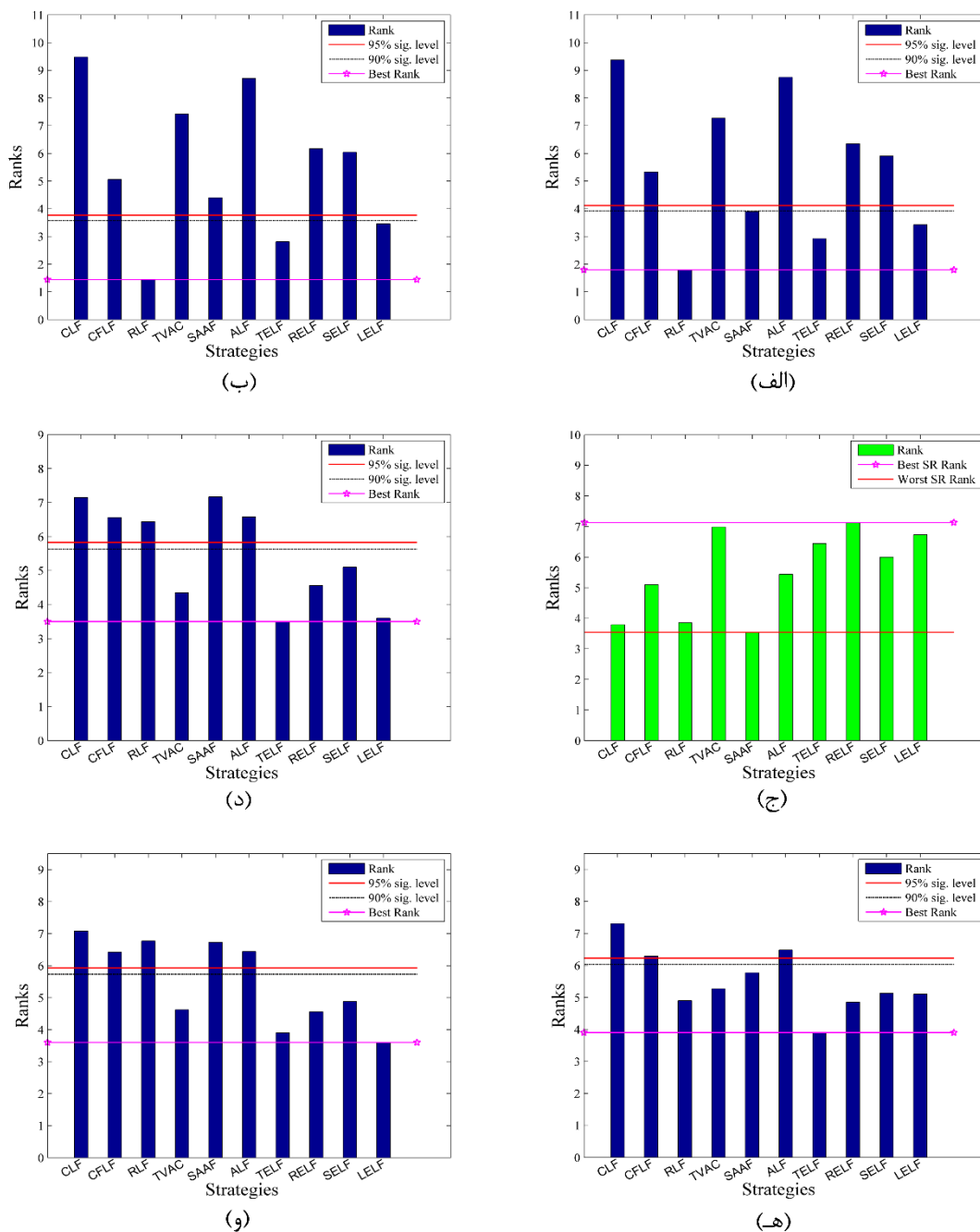
کمینه خطا						-----	
ALF	SAAF	TVAC	RLF	CFLF	CLF	نتیجه	الگوریتم
102	133	85	49	130	152	R^+	TELF
18	3	6	71	6	19	R^-	
0.017	0.001	0.006	0.532	0.001	0.004	p-value	
111	67	94	52	114	145	R^+	RELF
25	53	11	84	6	26	R^-	
0.026	0.691	0.009	0.408	0.002	0.010	p-value	
110	71	94	46	126	147	R^+	SELF
26	65	11	90	10	24	R^-	
0.030	0.877	0.009	0.255	0.003	0.007	p-value	
109	64.5	99	38	96	139	R^+	LELF
27	55.5	21	98	9	32	R^-	
0.034	0.798	0.027	0.121	0.006	0.020	p-value	

جدول ۱۲. نتایج آزمون فریدمن بر اساس جدول ۲ و جدول ۳.

SR	MNS	ANS	-----	-----
26	26	26	<i>N</i>	نتایج
64.7	171.7	162.8	Chi-Square	
1.7×10^{-10}	2.7×10^{-32}	1.9×10^{-30}	p-value	
3.79	9.48	9.37	CLF	رتبه‌بندی میانگین
5.10	5.06	5.33	CFLF	
3.85	1.44	1.79	RLF	
6.98	7.42	7.27	TVAC	
3.54	4.40	3.90	SAAF	
5.44	8.71	8.75	ALF	
6.44	2.81	2.92	TELF	
7.13	6.17	6.35	RELF	
6.00	6.04	5.90	SELF	
6.73	3.46	3.42	LELF	

جدول ۱۳. نتایج آزمون فریدمن بر اساس جدول ۴ تا جدول ۷.

STD	ME	AE	-----	-----
26	26	26	<i>N</i>	نتایج
46.4	39.1	58.7	Chi-Square	
5.0×10^{-7}	1.1×10^{-5}	2.4×10^{-9}	p-value	
7.08	7.31	7.15	CLF	رتبه‌بندی میانگین
6.42	6.29	6.56	CFLF	
6.77	4.90	6.44	RLF	
4.62	5.27	4.35	TVAC	
6.73	5.77	7.17	SAAF	
6.44	6.48	6.58	ALF	
3.9	3.90	3.50	TELF	
4.56	4.85	4.56	RELF	
4.88	5.13	5.10	SELF	
3.6	5.10	3.60	LELF	

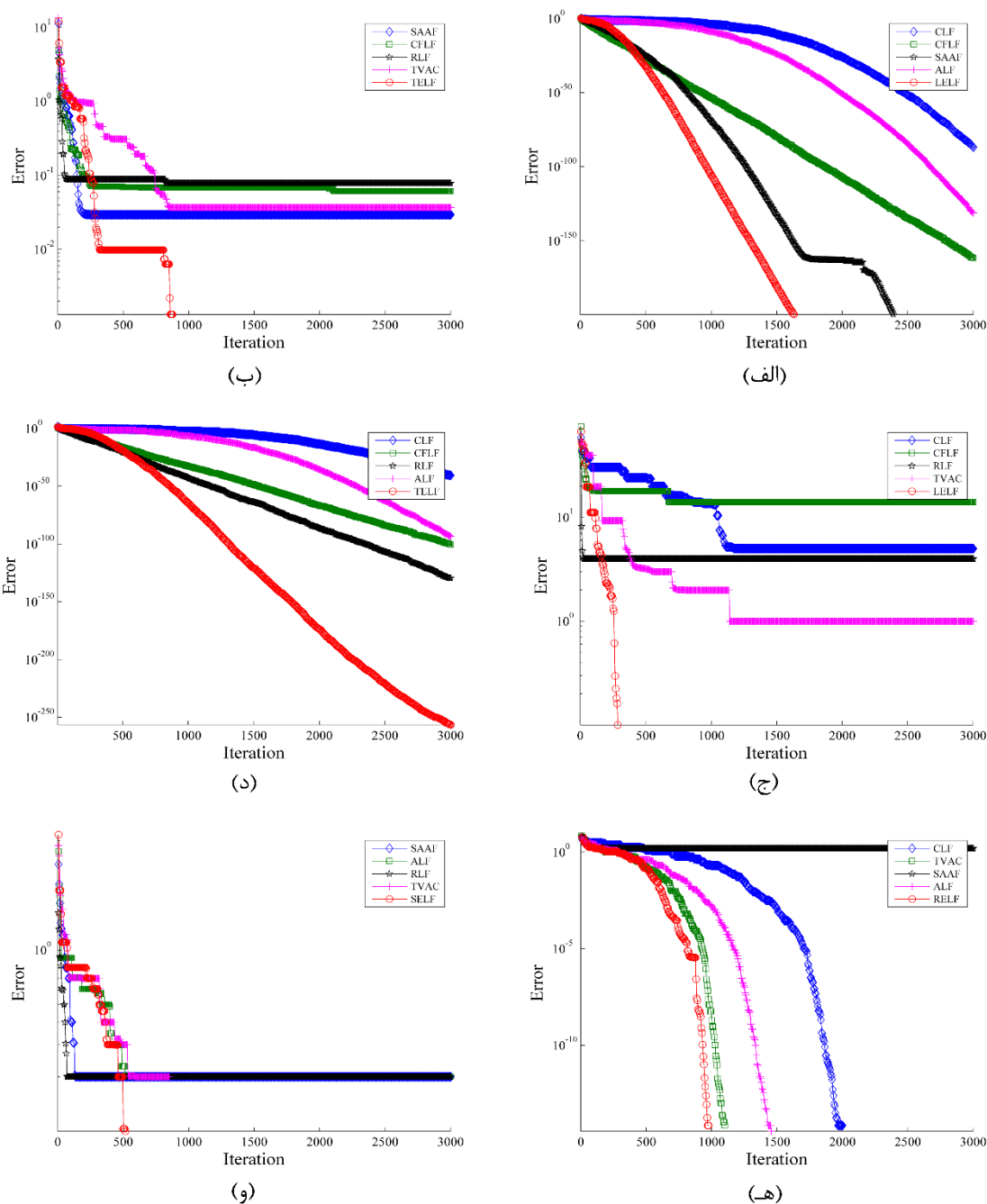


شکل ۲. نمودارهای میله‌ای بونفرونی-دان.

(الف) میانگین تعداد تکرارها مطابق جدول ۲ و جدول ۳. (ب) کمینه تعداد تکرارها مطابق جدول ۲ و جدول ۳.

(ج) نرخ موفقیت مطابق جدول ۲ و جدول ۳. (د) میانگین خطا مطابق جدول ۴ تا جدول ۷.

(هـ) کمینه خطا مطابق جدول ۴ تا جدول ۷. (و) انحراف معیار خطا مطابق جدول ۴ تا جدول ۷.



شکل ۳. گراف همگرایی مقایسه‌ای گونه‌های مختلف الگوریتم PSO.

(الف) تابع Sphere با $\varepsilon = 10^{-200}$. (ب) تابع Griewank با $\varepsilon = 10^{-3}$.

(ج) تابع Rastrigin با $\varepsilon = 10^{-1}$. (د) تابع Zakharov با $\varepsilon = 10^{-300}$.

(هـ) تابع Weierstrass با $\varepsilon = 10^{-20}$. (و) تابع Salomon با $\varepsilon = 10^{-1}$.

۷- نتیجه‌گیری

الگوریتم PSO یک تکنیک محاسباتی تصادفی بوده و برگرفته از رفتار اجتماعی موجودات در گروه‌های بزرگ است و برای مقاصد بهینه‌سازی، کاربرد دارد. رویکردهای مختلفی برای درک چگونگی کارکرد این الگوریتم و تلاش برای بهبود همگرایی در حل مسائل مختلف، مورد استفاده قرار گرفته است. این رویکردها از تحلیل‌های ریاضی تا آزمایش‌های عددی را شامل می‌شوند. اگرچه تاکنون جامعه علمی توانسته است انواع گوناگونی از مسائل مهندسی را حل کند، اما تنظیم پارامترهای الگوریتم PSO همچنان یکی از مشکلات اصلی آن است.

در این مقاله، یک الگوریتم PSO بهبودیافته، ساده، کارآمد، سریع و قابل اطمینان با یک کلاس نوین از استراتژی‌های ضرائب یادگیری نمایی (ELF) پیشنهاد شد و به‌طور موفقیت‌آمیزی برای حل مسائل بهینه‌سازی معیار به‌کار گرفته شد. نتایج آزمایشات ثابت کنند که تکنیک‌های پیشنهادی، عملکرد الگوریتم PSO را از لحاظ سرعت همگرایی و دقت جواب بهبود می‌بخشند. همچنین نتایج حاصل از روش پیشنهادی با برخی از معروف‌ترین گونه‌های الگوریتم PSO مقایسه شد. هرچند آزمون‌های آماری نتیجه دهند که الگوریتم‌های کلاس پیشنهادی ELF-PSO از نظر معیارهای مختلف ارزیابی عملکرد مانند میانگین خطا، کمینه خطا و انحراف معیار بر سایر الگوریتم‌های مشابه برتری دارند، اما طبق قضیه «هیچ ناهاری مجانی نیست»، یک الگوریتم بهینه‌سازی نمی‌تواند نسبت به بقیه الگوریتم‌ها از هر نظر یا روی هر نوع مسأله‌ای، عملکرد بهتری داشته باشد. به‌عبارت دیگر ممکن است روش‌های پیشنهادی، برای حل برخی از مسائل جدید نیاز به تغییر یا بهبود داشته باشند که می‌تواند موضوع مقالات بعدی باشد. علاوه بر این، کارهای آتی می‌تواند شامل اجرای الگوریتم‌های کلاس ELF-PSO برای حل مسائل بسیار پیچیده کنونی مهندسی از جمله بخش‌بندی تصاویر پزشکی در مهندسی پزشکی باشد.

مراجع

- [1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, Nagoya, Japan, 1995.
- [2] R. Eberhart and J. Kennedy, "Particle swarm optimization", Proceeding of IEEE International Conference on Neural Network, Perth, Australia, 1995.
- [3] S.K. Satapathy, S. Dehuri and A.K. Jagadev, "EEG signal classification using PSO trained RBF neural network for epilepsy identification", Informatics in Medicine Unlocked, 2017, 6 1-11.
- [4] Y. Zhang, E. Chen, P.-j. Guo and C. Ma, Application of improved particle swarm optimization BP neural network in transformer fault diagnosis, in: Chinese Automation Congress (CAC), 2017, IEEE, 2017, pp. 6971-6975.
- [5] M. Eslami, M.R. Shayesteh and M. Pourahmadi, "Optimal Design of PID-based Low-Pass Filter for Gas turbine Using Intelligent Method", IEEE Access, 2018.
- [6] T.X. Pham, P. Siarry and H. Oulhadj, "Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation", Applied Soft Computing, 2018, 65 230-242.
- [7] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease", Neurocomputing, 2018, 320 195-202.

- [8] N. Bharanidharan and H. Rajaguru, Performance analysis of particle swarm optimization technique in classification of dementia using MRI images, in: Communication and Electronics Systems (ICCES), 2017 2nd International Conference on, IEEE, 2017, pp. 311-315.
- [9] M. Sabeti, R. Boostani and B. Davoodi, "Improved particle swarm optimisation to estimate bone age", IET Image Processing, 2017, 12 (2), 179-187.
- [10] Z. Hu, D. Zou, Z. Kong and X. Shen, A Particle Swarm Optimization Algorithm with Time Varying Parameters, in: 2018 Chinese Control And Decision Conference (CCDC), IEEE, 2018, pp. 4555-4561.
- [11] D. Wu, Cloud Computing Task Scheduling Policy Based on Improved Particle Swarm Optimization, in: 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), IEEE, 2018, pp. 99-101.
- [12] J.J. Liang, A.K. Qin, P.N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Transactions on Evolutionary Computation, 2006, 10 (3), 281-295.
- [13] M.J. Amoshahy, M. Shamsi and M.H. Sedaaghi, "A novel flexible inertia weight particle swarm optimization algorithm", PloS one, 2016, 11 (8), e0161558.
- [14] H. Xing and X. Pan, Application of improved particle swarm optimization in system identification, in: 2018 Chinese Control And Decision Conference (CCDC), IEEE, 2018, pp. 1341-1346.
- [15] M. Merchaoui, A. Sakly and M.F. Mimouni, Improved fast particle swarm optimization based PV MPPT, in: Renewable Energy Congress (IREC), 2018 9th International, IEEE, 2018, pp. 1-7.
- [16] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization", Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, Washington, USA, 1999.
- [17] R.C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, CA, USA, 2000.
- [18] S. Agrawal, S. Silakari and J. Agrawal, "Adaptive Particle Swarm Optimizer with Varying Acceleration Coefficients for Finding the Most Stable Conformer of Small Molecules", Molecular informatics, 2015, 34 (11-12), 725-735.
- [19] S.U. Khan, S. Yang, L. Wang and L. Liu, "A modified particle swarm optimization algorithm for global optimizations of inverse problems", IEEE Transactions on Magnetics, 2016, 52 (3), 1-4.
- [20] Y. Shi and R. Eberhart, "A modified particle swarm optimizer", Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, Anchorage, USA, 1998.
- [21] H.-R. Li and Y.-L. Gao, "Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation", Information and Computing Science, 2009. ICIC'09. Second International Conference on, Manchester, UK, 2009.
- [22] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", IEEE Transactions on Evolutionary Computation, 2002, 6 (1), 58-73.

- [23] P.N. Suganthan, "Particle swarm optimiser with neighbourhood operator", Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, Washington, USA, 1999.
- [24] A. Ratnaweera, S.K. Halgamuge and H.C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on Evolutionary Computation, 2004, 8 (3), 240-255.
- [25] S.S. Jadon, H. Sharma, J.C. Bansal and R. Tiwari, "Self adaptive acceleration factor in particle swarm optimization", Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Gwalior, India, 2013.
- [26] D.H. Wolpert and W.G. Macready, "No free lunch theorems for optimization", IEEE Transactions on Evolutionary Computation, 1997, 1 (1), 67-82.
- [27] J. Derrac, S. García, D. Molina and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", Swarm and Evolutionary Computation, 2011, 1 (1), 3-18.