E-ISSN: 2676-6159 | P-ISSN: 2538-5079

Journal of Decisions and Operations Research

www.journal-dmor.ir

J. Decis. Oper. Res. Vol. 6, No. 3 (2021) 304–329.

Paper Type: Original Article

A New Meta-Heuristic Algorithm: Military Optimization Algorithm (MOA)

Hojatollah Rajabi Moshtaghi [1], Abbas Toloie Eshlaghy[1,*], Mohammad Reza Motadel[2]

[1] Department of Industrial Management, Science and Research Branch, Islamic Azad University, Tehran, Iran; hrajabi55@gmail.com; toloei@gmail.com;

[2] Department of Industrial Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran; dr.motadel@gmail.com.

Citation:

Rajabi Moshtaghi, H., Toloie Eshlaghy, A., & Motadel, M. R. (2021). A new meta-heuristic algorithm: military optimization algorithm (MOA). *Journal of decisions and operations research,* 6 (3), 304-329.

Received: 04/03/2021 Reviewed: 12/04/2021 Revised: 05/07/2021 Accept: 15/07/2021

Abstract

Purpose: In recent years, meta-heuristic algorithms and their application in solving complicated, nonlinear, and high dimensions problems have increased dramatically and the fact that meta-heuristic algorithms are used to solve complex and changing problems of real life, has caused the algorithms world and their design to be very dynamic and alive; that's why new algorithms are constantly being created. Hence, the purpose of this research is to introduce a novel meta-heuristic algorithm called Military Optimization Algorithm (MOA).

Methodology: Inspired by military operations, the proposed algorithm was designed and presented. After coding, Standard test functions and benchmark algorithms were determined to evaluate the performance of the algorithm.

Findings: The performance of new algorithm is analyzed by 23 standard test functions and compared to 8 benchmark meta-heuristic algorithms including: Genetic Algorithm, Particle Swarm Optimization, Artificial Bee Colony, Shuffled Frog Leaping Algorithm, and Imperialist Competitive Algorithm, Grey Wolf Optimizer, Whale Optimization Algorithm, and Grasshopper Optimization Algorithm, by considering three indices of "average answers", "time complexity of algorithm (speed)" and "Convergence speed/ time". The results show the excellent performance of the proposed algorithm.

Originality/Value: In this paper, inspired by military operations, a novel meta-heuristic algorithm called MOA is introduced. It is population-based and stable with "random search", "dividing solution space into several regions and allocating a part of the population to each region", "cavalry search", and "infantry search".

Keywords: Optimization, Meta-Heuristic algorithms, Military optimization algorithm, Evolutionary algorithms, Swarm algorithms.

Corresponding Author: toloei@gmail.com

10.22105/dmor.2021.276125.1333

Licensee. **Journal of Decisions and Operations Research**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

Journal of Decisions and Operations Research

تصمیم‌گیری و تحقیق در عملیات

دوره ۶، شماره ۳، (۱۴۰۰)، ۳۲۹-۳۰۴

www.journal-dmor.ir

نوع مقاله: پژوهشی

ارائه یک الگوریتم فرا ابتکاری جدید: الگوریتم بهینه-سازی نظامی

حجت اله رجبی مشتاقی[۱]، عباس طلوعی اشلقی[۱،*]، محمدرضا معتدل[۲]

[۱]گروه مدیریت صنعتی، دانشکده مدیریت و اقتصاد، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران.

[۲]گروه مدیریت صنعتی، دانشکده مدیریت، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران.

چکیده

هدف: در سال‌های اخیر، شاهد ظهور و گسترش الگوریتم‌های فرا ابتکاری و استفاده از آن‌ها جهت حل مسائل پیچیده، غیرخطی و ابعاد بالا بوده‌ایم. با توجه به اینکه الگوریتم‌های فوق برای حل مسائل پیچیده و در حال تغییر دنیای واقعی به کار می‌روند، دنیای الگوریتم‌ها و طراحی آن‌ها به شکل فزاینده‌ای پویا و رو به رشد بوده است. بنابراین، پیوسته شاهد به وجود آمدن الگوریتم‌های جدیدی هستیم. هدف از این تحقیق، ارائه یک الگوریتم فرا ابتکاری جدید به نام «الگوریتم بهینه‌سازی نظامی» می‌باشد.

روش‌شناسی پژوهش: با الهام از عملیات‌های نظامی الگوریتم پیشنهادی طراحی و ارائه گردید و پس از کدنویسی، توابع تست استاندارد و الگوریتم‌های محک برای ارزیابی عملکرد آن تعیین و مشخص شدند.

یافته‌ها: عملکرد الگوریتم پیشنهادی به‌وسیله ۲۳ تابع تست استاندارد و با در نظر گرفتن شاخص‌های «میانگین جواب‌ها»، «میانگین زمان محاسباتی» و «زمان همگرایی» در مقایسه با هشت الگوریتم محک شامل: ژنتیک، ازدحام ذرات، کلونی زنبور مصنوعی، قورباغه جهنده، رقابت استعماری، گرگ خاکستری، بهینه‌سازی وال و بهینه‌سازی ملخ مورد ارزیابی و سنجش قرار گرفت. نتایج نشان‌دهنده عملکرد مطلوب الگوریتم پیشنهادی است.

اصالت/ارزش افزوده علمی: در این مقاله، با الهام از عملیات‌های نظامی الگوریتم جدیدی به نام الگوریتم بهینه‌سازی نظامی (*MOA*) ارائه می‌شود که مبتنی بر جمعیت است و بر اساس «جستجوی تصادفی»، «تقسیم فضای جواب به چند منطقه و تخصیص بخشی از جمعیت به هر منطقه»، «جستجوی سواره‌نظام» و «جستجوی پیاده‌نظام» عمل می‌کند.

کلیدواژه‌ها: بهینه‌سازی، الگوریتم‌های فراابتکاری، الگوریتم بهینه‌سازی نظامی، الگوریتم‌های تکاملی، الگوریتم‌های ازدحامی.

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ۶، شماره ۳، پاییز ۱۴۰۰، صفحه: ۳۲۹-۳۰۴

۱- مقدمه

بشر همواره در تلاش برای رسیدن به کمال (حالت بهینه) در جنبه‌های گوناگون زندگی خود بوده است و لذا یکی از اساسی‌ترین مسائل موجود در جهان جستجو برای رسیدن به همین موضوع است (صفوی و همکاران[۱]، ۲۰۱۴). در دنیای واقعی مسائل زیادی از قبیل حمل‌ونقل، جایابی انبارها یا نقاط فروش محصولات، طراحی شبکه‌های ارتباطی و زمان‌بندی همگی از مسائل پیچیده و ترکیبی به شمار می‌روند. این مسائل در اندازه‌های عملی بزرگ‌اند و نمی‌توان جواب بهینه آن‌ها را در زمان قابل قبولی به دست آورد. نیاز به حل این مسائل، چاره دیگری جز بسنده کردن به جواب‌های نزدیک بهینه باقی نگذارده است، البته با این شرط که راه‌حل‌ها قابل‌پذیرش باشند و در زمان معقولی حاصل شوند. بدین

[1] Safavi et al.

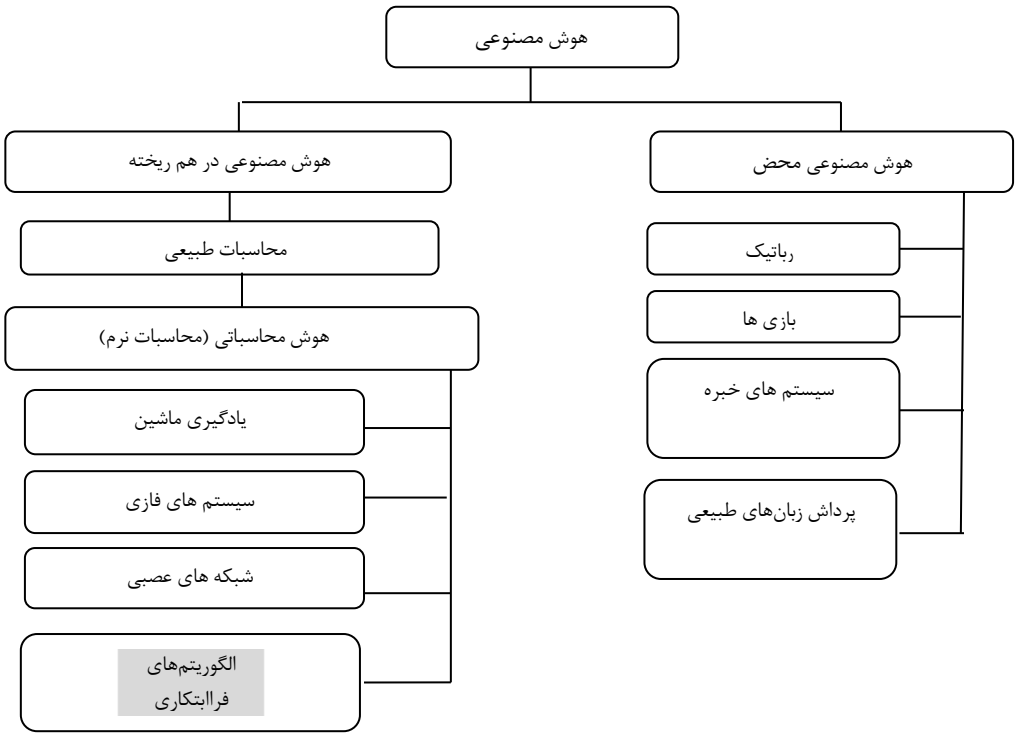
* نویسنده مسئول

toloie@gmail.com

10.22105/dmor.2021.276125.1333

منظور الگوریتم‌های «ابتکاری[1]» ارائه شدند که می‌توانند تااندازه‌ای مناسب عمل کرده و گاهی تا نزدیک بهینگی پیش روند. بااین‌حال، یکی از مشکلات حل‌نشده آن‌ها قرار گرفتن در بهینه‌های محلی و ناکامی در ارائه پاسخ صحیح بوده است. در پاسخ به این محدودیت، الگوریتم‌های «فرا ابتکاری[2]» توسعه داده شدند، هرچند آن‌ها نیز همچنان و البته با درجات کمتر دچار مشکل بهینه‌های محلی‌اند (عالم تبریز و همکاران[3]، ۲۰۰۸). این الگوریتم‌ها روشی برای حل مسائل بهینه‌سازی‌اند که با ایجاد جواب (های) تصادفی و بر اساس عملگرهای خود، با تغییر جواب (های) تصادفی ایجادشده به سمت بهینگی حرکت می‌کنند (قهرمانی نهر[4]، ۲۰۱۹). در سال‌های اخیر استفاده از الگوریتم‌های فرا ابتکاری برای حل مسائل بهینه‌سازی گسترش قابل‌توجهی یافته است و امروزه کاربردهای وسیعی در شاخه-های مختلف علوم دارد. در حل این‌گونه مسائل، الگوریتم‌های فرا ابتکاری می‌توانند، جواب مناسب را درزمانی قابل‌قبول محاسبه کنند (یقینی و اخوان کاظم‌زاده[5]، ۲۰۱۶).

الگوریتم‌های فرا ابتکاری زیرشاخه‌ای از هوش محاسباتی یا محاسبات نرم‌اند که خود زیرشاخه‌ای از هوش مصنوعی است. **شکل ۱** جایگاه الگوریتم‌های فرا ابتکاری را در هوش مصنوعی نشان می‌دهد (محمدپورزرندی[6]، ۲۰۱۳).



شکل ۱- جایگاه الگوریتم‌های فرا ابتکاری در هوش مصنوعی.

Figure 1- The place of meta-heuristic algorithms in artificial intelligence.

به‌صورت کلی عملکرد الگوریتم‌های فرا ابتکاری برای به دست آوردن جواب بهینه، تااندازه‌ای شبیه هم است. در اغلب این الگوریتم‌ها، جستجو با ایجاد یک یا تعدادی جواب تصادفی (با توجه به نوع الگوریتم ازلحاظ مبتنی بودن بر یک جواب[7] و یا جمعیتی از جواب‌ها[8]) در فضای جواب آغاز می‌شود. به جواب‌های اولیه ایجادشده (در الگوریتم‌های مبتنی بر جمعیت) جمعیت، کلونی، گروه و همچنین به هر یک از جواب‌ها به‌تنهایی، کروموزوم، ذره، مورچه و غیره گفته می‌شود.سپس با استفاده از عملگرها جواب یا جواب‌های جدید تولید می‌شود. سپس با شیوه‌های مختلف، جواب‌هایی از میان جواب‌های قبلی و جدید انتخاب می‌شود و عملیات تا حصول معیار توقف ادامه می‌یابد (شریف‌زاده و امجدی[9]، ۲۰۱۴). در سال‌های اخیر و به‌ویژه پس از سال ۲۰۰۰، الگوریتم‌های فرا ابتکاری رشد و توسعه

[1] Heuristic
[2] Meta-heuristic
[3] Alam Tabriz et al.
[4] Ghahramani Nahr
[5] Yaghini and Akhavan Kazemzadeh
[6] Mohammad Pour Zarandi
[7] Single-solution
[8] Population-based
[9] Sharifzadeh and Amjady

فزاینده‌ای داشته‌اند. در **جدول ۱** تعدادی از الگوریتم‌های فرا ابتکاری با ذکر مشخصات آن‌ها از قبیل نوع الگوریتم (تکاملی، ازدحامی، مبتنی بر فیزیک و مبتنی بر انسان)، منبع الهام، الهام گرفته‌شده از طبیعت در برابر غیر الهامی از طبیعت و سال ارائه، به همراه توضیحی مختصر درباره هر یک آورده شده‌اند.

جدول ۱- لیست الگوریتم‌های فرا ابتکاری به همراه مشخصات و ویژگی‌های آن‌ها.

Table 1- List of meta-heuristic algorithms with their specifications and features

ردیف	نام الگوریتم	نوع الگوریتم	منبع الهام	الهام گرفته شده از طبیعت در برابر غیر الهامی از طبیعت	سال	شرح مختصر	مرجع
۱	الگوریتم ژنتیک (GA)[۱]	تکاملی	زیست و بیولوژی	طبیعت	۱۹۷۵	از مفاهیم علم زیست شناسی مانند وراثت و جهش استفاده می‌کند.	(هلند[۲]، ۱۹۷۵)
۲	الگوریتم شبیه‌سازی تبرید[۳](SA)	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۱۹۸۳	با الهام از فرایند ذوب و انجماد فلزات در ابعاد مولکولی پیشنهاد شده‌است.	(کریک پاتریک و همکاران[۴]، ۱۹۸۳)
۳	الگوریتم دسته پرندگان[۵] (PSO)	ازدحامی	حیوانات	طبیعت	۱۹۹۵	از پرواز گروهی پرندگان الهام گرفته شده است و دارای دو اپراتور می باشد: اپراتور به روز رسانی سرعت و اپراتور به روز رسانی موقعیت.	(کندی و ایبرهارت[۶]، ۱۹۹۵)
۴	الگوریتم مورچگان[۷] (ACO)	ازدحامی	حیوانات	طبیعت	۱۹۹۶	بر مبنای کاوش در رفتار مورچه‌های واقعی است.	(دوریگو و همکاران[۸]، ۱۹۹۶)
۵	الگوریتم قورباغه جهنده[۹] (SFLA)	تکاملی	حیوانات	طبیعت	۲۰۰۳	اساس کار SFLA بر شبیه‌سازي رفتار قورباغه‌ها براي يافتن غذا در تالاب پايه‌ريزي شده است.	(ایساف و لنزی[۱۰]، ۲۰۰۳)

[1] Genetic Algorithm (GA)
[2] Holland
[3] Simulated Annealing (SA)
[4] Kirkpatrick et al.

[5] Particle Swarm Optimization (PSO)
[6] Kennedy and Eberhart
[7] Ant Colony Optimization (ACO)
[8] Dorigo et al.
[9] Shuffled Frog Leaping Algorithm (SFLA)
[10] Eusuff and Lansey

جدول ۱- (ادامه).

Table 1- (Continued).

۶	الگوریتم رقابت استعماری[۱] (ICA)	تکاملی	انسان	غیرطبیعت	۲۰۰۷	اعضای جمعیت به عنوان کشور در نظر گرفته می‌شوند. کشورها به دو گروه استعمارگر و مستعمره (کلونی) تقسیم و امپراتوری‌ها شکل می‌گیرند.	(آتش‌پز گرگری و لوکاس[۲]، ۲۰۰۷)
۷	الگوريتم كلوني زنبور مصنوعي[۳] (ABC)	ازدحامی	حیوانات	طبیعت	۲۰۰۷	در این الگوریتم از رفتار گروهی زنبور عسل در کشف غذا استفاده شده و عملکرد زنبورهای کارگر، مراقب و پیشاهنگ مدل‌سازی شده است.	(کارابوگا و باستورک[۴]، ۲۰۰۷)
۸	بهینه‌سازی مبتنی بر جغرافیای زیستی (BBO)[۵]	تکاملی	زیست و بیولوژی	طبیعت	۲۰۰۸	الهام گرفته شده از توزیع جغرافیایی ارگانیسم‌های بیولوژیکی بوده و از عملگرهای پایه‌ایی همانند مهاجرت و جهش استفاده می‌کند.	(سایمون[۶]، ۲۰۰۸)
۹	الگوریتم الهام گرفته شده از انسان (HIA)[۷]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۰۹	فضای جستجوی را به صورت مساوی به چند فضای فرعی تقسیم کرده و به تعداد مساوی عامل جستجو به هر یک از فضاهای فرعی تخصیصی می‌دهد.	(ژانگ و همکاران[۸]، ۲۰۰۹)
۱۰	الگوریتم جستجوی مبتنی بر راهنمایی مشاور (CGS)[۹]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۱۰	براساس تبادل مستقیم اطلاعات بین افراد یک جمعیت ارائه شده است و از روش تصمیم‌گیری افراد در دنیای واقعی برمبنای راهنمایی گرفتن از مشاوران، الهام گرفته شده است.	(اورداچ[۱۰]، ۲۰۱۰)
۱۱	بهینه‌سازی واکنش شیمیایی(CRO)[۱۱]	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۲۰۱۰	این الگوریتم از رفتار مولکول‌ها در واکنش‌های شیمیایی و تبادل انرژی در آن‌ها الهام گرفته شده است.	(لم و لی[۱۲]، ۲۰۱۰)
۱۲	الگوريتم مبتني بر كهكشان(GbSA)[۱۳]	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۲۰۱۱	فضاي جواب را براي يافتن يك جواب بهتر با استفاده بازوي مارپيچي مانند جستجو مي‌كند و براي فرار از بهينه محلي حركت مارپيچي به‌وسيله آشوب / بي نظمي بهبود مي‌يابد.	(شاه حسینی[۱۴]، ۲۰۱۱)

[1] Imperialist Competitive Algorithm (ICA)
[2] Atashpaz-Gargari and Lucas
[3] Artificial Bee Colony Algorithm (ABC)
[4] Karaboga and Basturk
[5] Biogeography Based Optimization (BBO)
[6] Simon
[7] Human-Inspired Algorithm (HIA)
[8] Zhang et al.
[9] Consultant-Guided Search (CGS)
[10] Iordache
[11] Chemical Reaction Optimization (CRO)
[12] Lam and Li
[13] Galaxy based Search Algorithm (GbSA)
[14] Shah-Hosseini

جدول ۱- (ادامه).

Table 1- (Continued).

۱۳	بهینه‌سازی مبتنی بر آموزش-یادگیری (TLBO)[۱]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۱۱	در این روش فرایند جستجو شامل دو فاز می‌باشد: در فاز اول یادگیری به‌وسیله تاثیر معلم انجام می‌شود، در حالی‌که در فاز دوم توسط تعامل دوجانبه.	(رائو و همکاران[۲]، ۲۰۱۱)
۱۴	بهینه‌سازی جامعه هرج و مرج (ASO)[۳]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۱۲	این الگوریتم، مسأله را در قالب یک گروه با افراد آشوب طلب بی‌ثبات، حادثه‌جو و متنفر از محیط آرام، برای غلبه بر گیر افتادن در بهینه محلی مدل‌سازی می کند.	(شایقی و داداشپور[۴]، ۲۰۱۲)
۱۵	بهینه‌ساز گرگ خاکستری(GWO)[۵]	ازدحامی	حیوانات	طبیعت	۲۰۱۴	از مکانیزم شکار و سلسله مراتب رهبري گرگ‌هاي خاكستري تقليد مي‌كند.	(میر جلیلی و همکاران[۶]، ۲۰۱۴)
۱۶	توپ طلايي(GB)[۷]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۱۴	اين الگوريتم مبتني بر جمعيت چندگانه بوده و بر مفاهيم بازي فوتبال متكی مي‌باشد.	(اوسابا و همکاران[۸]، ۲۰۱۴)
۱۷	الگوریتم تبادل بازار (EMA)[۹]	تکاملی	انسان	غیرطبیعت	۲۰۱۴	مبتنی بر مکانیسم معاملات سهام در بورس کالا می‌باشد.	(قربانی و بابایی[۱۰]، ۲۰۱۴)
۱۸	الگوریتم حرکت یونی (IMA)[۱۱]	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۲۰۱۵	با الهام از حركت يوني و نيروي كشش و فشار آنيون‌ها و كاتيون‌ها ارائه شده است.	(جاویدی و همکاران[۱۲]، ۲۰۱۵)
۱۹	جستجوی فراکتال استوکستیک (SFS)[۱۳]	تکاملی	سایر	طبیعت	۲۰۱۵	با الهام از پديه طبيعي رشد ارائه شده است كه از يك مفهوم رياضي به نام «فراكتال[۱۴]»استفاده مي‌كند.	(سلیمی[۱۵]، ۲۰۱۵)
۲۰	بهینه‌سازی طوفان مغزی (BSO)[۱۶]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۱۵	این الگوریتم از فرایند طوفان مغزی برای حل مسأله الهام گرفته شده است.	(شی[۱۷]، ۲۰۱۵)

[12] Teaching-Learning Based Optimization (TLBO)
[2] Rao et al.
[3] Anarchic Society Optimization (ASO)
[4] Shayeghi and Dadashpour
[5] Grey Wolf Optimizer (GWO)
[6] Mirjalili et al.
[7] Golden Ball (GB)
[8] Osaba et al.
[9] Exchange Market Algorithm (EMA)
[10] Ghorbani and Babaei
[11] Ions Motion Algorithm (IMA)
[12] Javidy et al.
[13] Stochastic Fractal Search (SFS)
[14] Fractal
[15] Salimi
[16] Brain Storm Optimization (BSO)
[17] Shi

جدول ۱- (ادامه).

Table 1- (Continued).

۲۱	الگوریتم بهینه‌سازی شیر (LOA)[1]	ازدحامی	حیوانات	طبیعت	۲۰۱۶	در این روش شیرهای ساکن به عنوان عامل جستجوی محلی مدل شده‌اند، در حالی‌که شیرهای کوچ نشین به عنوان عامل‌های جستجوی سراسری برای شناسایی و اکتشاف مدل گردیده‌اند.	(یزدانی و جولای[2]، ۲۰۱۶)
۲۲	الگوریتم بهینه‌سازي وال(WOA)[3]	ازدحامی	حیوانات	طبیعت	۲۰۱۶	این الگوریتم دارای ۳ گام می باشد: محاصره شکار، حمله به شکار (گام استخراج) و جستجوی شکار (گام شناسایی / اکتشاف).	(میر جلیلی و لوئیس[4]، ۲۰۱۶)
۲۳	الگوریتم سنجاقک (DA)[5]	ازدحامی	حیوانات	طبیعت	۲۰۱۶	از رفتار اجتماعی سنجاقک‌ها در زمان جستجوی غذا، هدایت و دوری از دشمن در یک گروه الهام گرفته شده‌است.	(میر جلیلی[6]، ۲۰۱۶)
۲۴	الگوریتم جستجوی الکترو (ESA)[7]	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۲۰۱۷	این الگوریتم از حرکت الکترون‌ها در اطراف هسته یک اتم الهام گرفته شده است.	(طبری و احمد[8]، ۲۰۱۷)
۲۵	الگوریتم بهینه‌سازي تبادل حرارتي (TEOA)[9]	مبتنی بر فیزیک و شیمی	فیزیک و شیمی	طبیعت	۲۰۱۷	این الگوریتم بر اساس قانون خنك‌سازي نيوتن[10]، ارائه شده‌است که از نظر تئوريكي و كاربردي نسبتا" ساده مي‌باشد.	(کاوه و دادرس[11]، ۲۰۱۷)
۲۶	الگوریتم بهینه‌سازي ازدحام سوسک (BSO)[12]	ازدحامی	حیوانات	طبیعت	۲۰۱۸	این الگوریتم براساس ارتقاء عملکرد بهینه‌سازی ازدحامی از طریق اصول کاوش سوسک پیشنهاد شده است.	(وانگ و یانگ[13]، ۲۰۱۸)
۲۷	الگوریتم ازدحامی دلفین کوانتومی (QDSA)[14]	ازدحامی	حیوانات	طبیعت	۲۰۱۹	در این روش از الگوریتم جستجوی کوانتوم برای فرار از بهینه محلی در الگوریتم ازدحام دلفین استفاده شده است.	(کیائو و یانگ[15]، ۲۰۱۹)

[1] Lion Optimization Algorithm (LOA)
[2] Yazdani and Jolai
[3] Whale Optimization Algorithm (WOA)
[4] Mirjalili and Lewis
[5] Dragonfly Algorithm (DA)
[6] Mirjalili
[7] Electro-Search Algorithm (ESA)
[8] Tabari and Ahmad
[9] Thermal Exchange Optimization (TEO)
[10] Newton's Law of Cooling
[11] Kaveh and Dadras
[12] Beetle Swarm Optimization (BSO)
[13] Wang and Yang
[14] Quantum Dolphin Swarm Algorithm (QDSA)
[15] Qiao and Yang

جدول ۱- (ادامه).

Table 1- (Continued).

۲۸	بهینه‌ساز بازی دارت (DGO)[۱]	مبتنی بر انسان	انسان	غیرطبیعت	۲۰۲۰	در این الگوریتم قواعد بازی دارت برای حل مسائل بهینه‌سازی، شبیه‌سازی می-شود.	(دهقانی و همکاران[۲]، ۲۰۲۰)
۲۹	الگوریتم گوزن قرمز (RDA)[۳]	ازدحامی	حیوانات	طبیعت	۲۰۲۰	این الگوریتم با تقلید از رفتار گوزن قرمز اسکاتلندی (مخصوصاً در فصل تولید مثل) ارائه شده است.	(فتح اللهی فرد و همکاران[۴]، ۲۰۲۰)
۳۰	الگوریتم بهینه‌سازي حسابی (AOA)[۵]	سایر	ریاضیات	غیرطبیعت	۲۰۲۱	این الگوریتم از رفتار توزیع عملگرهای اصلی حساب در ریاضیات از قبیل: جمع، تفریق، ضرب و تقسیم استفاده می-کند.	(ابوعلی‌گاه و همکاران[۶]، ۲۰۲۱)

.

[1] Darts Game Optimizer (DGO)
[2] Dehghani et al.
[3] Red Deer Algorithm (RDA)
[4] Fathollahi-Fard et al.
[5] Arithmetic Optimization Algorithm (AOA)
[6] Abualigah et al.

همانطوری‌که در **جدول ۱** قابل‌مشاهده است، اکثر الگوریتم‌های فرا ابتکاری از پدیده‌های طبیعی الهام گرفته‌شده و در دو دهه اخیر ارائه‌شده‌اند. هم‌چنین این الگوریتم‌ها به‌صورت کلی به چهار دسته قابل‌تقسیم می‌باشند: «تکاملی»، «ازدحامی»، «مبتنی بر فیزیک» و «مبتنی بر انسان».

قابلیت استفاده از الگوریتم‌های فرا ابتکاری برای حل مسائل پیچیده و در حال تغییر دنیای واقعی موجب شد تا به‌کارگیری الگوریتم‌ها و طراحی آن‌ها به شکلی پویا دنبال شود. هم‌چنین طبق نظریه *NFL*[1] منطقاً اثبات‌شده است هیچ الگوریتم فرا ابتکاری وجود ندارد که برای حل تمام مسائل بهینه‌سازی کارا باشد (ولپرت و مکریدی[2]، ۱۹۹۷). به همین دلیل است پیوسته شاهد ارائه الگوریتم‌های جدیدی هستیم. به‌طوری‌که از سال ۲۰۰۰ میلادی تاکنون الگوریتم‌های فرا ابتکاری زیادی ارائه‌شده‌اند. بنابراین در این مقاله، با الهام از عملیات‌های نظامی (جنگ)، یک الگوریتم فرا ابتکاری جدیدی به نام الگوریتم بهینه‌سازی نظامی[3] (*MOA*) پیشنهادشده است. سپس عملکرد آن به‌وسیله ۲۳ تابع تست[4] استاندارد در مقایسه با هشت الگوریتم فرا ابتکاری محک شامل: الگوریتم ژنتیک (*GA*)، بهینه‌سازی ازدحام ذرات (*PSO*)، کلونی زنبور مصنوعی (*ABC*)، الگوریتم قورباغه جهنده (*SFLA*)، رقابت استعماری (*ICA*)، بهینه‌سازی گرگ خاکستری (*GWO*)، بهینه‌سازی وال (*WOA*) و بهینه‌سازی ملخ (*GOA*) مورد ارزیابی قرار گرفت.

مرور مقالات و تحقیقات پیشین نشان می‌دهد که در بسیاری از این مطالعات ازجمله: ارائه الگوریتم «جستجوی فراکتال استوکستیک» (سلیمی، ۲۰۱۵). «کلونی زنبور مصنوعی» (کارابوگا و باستورک، ۲۰۰۷). «الگوریتم الهام گرفته‌شده از خفاش» (یانگ[5]، ۲۰۱۰). «بهینه‌ساز گرگ خاکستری» (میر جلیلی و همکاران، ۲۰۱۴). «الگوریتم بهینه‌سازی شیر» (یزدانی و جولای، ۲۰۱۶). «الگوریتم بهینه سازی وال» (میر جلیلی و لوئیس، ۲۰۱۶). «الگوریتم سنجاقک» (میر جلیلی، ۲۰۱۶) و «بهینه‌سازی مهاجرت حیوانات» (لی و همکاران[6]، ۲۰۱۴). بهینه‌سازی گردو بازی (دهقانی و همکاران[7]، ۲۰۲۰). صرفاً میانگین جواب‌های به‌دست‌آمده از هر الگوریتم موردبررسی قرارگرفته و به شاخص زمان اجرای الگوریتم (سرعت) توجه نشده است. در این مقاله، برای ارزیابی عملکرد الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها، علاوه بر توجه به هر دو شاخص میانگین جواب‌ها و زمان اجرا، شاخص سرعت / زمان همگرایی نیز موردتوجه قرارگرفته است.

بنابراین نوآوری‌های تحقیق حاضر را می‌توان به این شکل خلاصه کرد:

- *ارائه یک الگوریتم فرا ابتکاری جدید با عملکرد بهتر نسبت به سایر الگوریتم‌ها (محک) در یافتن جواب بهینه در برخی مسائل.*
- *سرعت بالای الگوریتم پیشنهادی نسبت به سایر الگوریتم‌ها (محک).*
- *توجه هم‌زمان به شاخص‌های میانگین جواب‌ها، زمان اجرای الگوریتم (سرعت) و زمان همگرایی.*

۲- الگوریتم بهینه‌سازی نظامی (MOA)

این الگوریتم با الهام از عملیات‌های نظامی پیشنهادشده است. در یک عملیات نظامی منطقه عملیات مشخص بوده و از عوامل، روش‌ها و مکانیزم‌های خاصی از قبیل سربازها، فرمانده / فرماندهان، پیاده‌نظام، سواره‌نظام، دسته‌ها / گردان‌ها، توپخانه و غیره برای اجرای عملیات استفاده می‌شود. در این الگوریتم از مفاهیم و روش‌های فوق تقلیدشده و برای جستجو و به دست آوردن جواب بهینه استفاده‌شده است. مراحل الگوریتم بهینه‌سازی نظامی به شرح زیر می‌باشد:

مرحله ۱: جواب‌های تصادفی.

[1] No Free Lunch
[2] Wolpert and Macready
[3] Military Optimization Algorithm
[4] Test Function
[5] Yang
[6] Li et al.
[7] Dehghani et al.

ایجاد تعداد M جواب تصادفی (M عدد نسبتاً بزرگی بوده و چند/چندین برابر تعداد جمعیت می‌باشد) X_m^R, $m=1, 2,...,M$ و محاسبه تابع هدف/تابع تناسب و تعیین بهترین جواب این مرحله (X_{best}^R).

مرحله ۲. تقسیم‌بندی فضای جواب و گروه‌بندی جمعیت.

تقسیم منطقه (فضای جواب) به ٤ منطقه به‌صورت مساوی و سپس ایجاد جمعیت اولیه برای هر یک از مناطق، به‌طوری‌که جمعیت هر منطقه برابر با یک‌چهارم کل جمعیت است. به‌عبارت‌دیگر، اگر N جمعیت کل باشد، k برابر است با یک‌چهارم N, X_i^{gp} . $p=1,2,3,4,$ $i=1,2,...,k$ و $X_1^{gp} = X_{best}^R$ به شرطی که در فضای جواب قرار داشته باشد.

محاسبه تابع هدف/تابع تناسب برای هر گروه و تعیین بهترین جواب هر گروه به‌عنوان فرمانده آن گروه (X_c^{gp}).

حرکت اعضا به سمت فرمانده (X_c^{gp}) و سپس بروز رسانی تابع هدف/تابع تناسب و فرمانده (X_c^{gp})، بر اساس **رابطه (۱)**؛

$$X_{i,new}^{gp} = X_{i,old}^{gp} + t_g.\alpha.rand(X_c^{gp} - X_i^{gp}), \qquad (۱)$$

تکرار بند قبل تا حصول معیار توقف (تعداد تکرار مشخص برای این مرحله) و بروز آوری بهترین جواب هر گروه.

اعمال معیار توقف و تعیین بهترین جواب هر گروه (X_{best}^{gp}).

مرحله ۳. سواره‌نظام (حرکت با اندازه گام‌های بزرگ).

ایجاد جمعیت اولیه سواره، X_j^H ، $j=1,2,...,N$، به‌طوری‌که $X_{best}^R = X_1^H$ ، $X_{best}^{g1} = X_2^H$ ، $X_{best}^{g2} = X_3^H$ ، $X_{best}^{g3} = X_4^H$ ، $X_{best}^{g4} = X_5^H$.

محاسبه تابع هدف/تابع تناسب برای هر یک از اعضا و تعیین بهترین جواب به‌عنوان فرمانده (X_c^H).

حرکت اعضا به سمت فرمانده (X_c^H) و سپس بروز رسانی تابع هدف/تابع تناسب و فرمانده (X_c^H)، بر اساس **رابطه (۲)**؛

$$X_{j,new}^H = X_{j,old}^H + t_H.\beta.rand(X_c^H - X_j^H), \qquad (۲)$$

تکرار بند قبل تا حصول معیار توقف و بروز آوری بهترین جواب.

اعمال معیار توقف و تعیین بهترین جواب (X_{best}^H) در این مرحله.

مرحله ۴. پیاده‌نظام (حرکت با اندازه گام‌های کوچک).

ایجاد جمعیت اولیه پیاده، X_j^I ، $j=1,2,... N,$ ، به‌طوری‌که $X_{best}^R = X_1^I$ ، $X_{best}^{g1} = X_2^I$ ، $X_{best}^{g2} = X_3^I$ ، $X_{best}^{g3} = X_4^I$ ، $X_{best}^{g4} = X_5^I$ ، $X_{best}^H = X_6^I$.

محاسبه تابع هدف/تابع تناسب برای هر یک از اعضا و تعیین بهترین جواب به‌عنوان فرمانده (X_c^I).

حرکت اعضا به سمت فرمانده (X_c^I) و سپس بروز رسانی تابع هدف/تابع تناسب و فرمانده (X_c^I)، بر اساس **رابطه (۳)**؛

$$X_{j,new}^I = X_{j,old}^I + t_1.\gamma.rand(X_c^I - X_j^I), \qquad (۳)$$

تکرار بند قبل تا حصول معیار توقف و بروز آوری بهترین جواب.

اعمال معیار توقف و تعیین بهترین جواب (X_{best}^I) در این مرحله به‌عنوان جواب نهایی مساله.

۱-۲- تعریف علائم و متغیرها

M: عدد نسبتاً بزرگی بوده و چند / چندین برابر تعداد جمعیت است.

N: تعداد جمعیت.

K: تعداد جمعیت هر گروه، *p=4*: تعداد گروه‌ها، $X_c^{...}$: فرمانده.

X_{best}^R : بهترین جواب به‌دست‌آمده در مرحله یک (مرحله پاسخ‌های تصادفی).

X_{best}^{gp} : ۴ و p=۱،۲،۳ : بهترین جواب‌های به‌دست‌آمده در مرحله دو.

X_{best}^H : بهترین جواب‌های به‌دست‌آمده در مرحله سه (سواره‌نظام).

X_{best}^I : بهترین جواب‌های به‌دست‌آمده در مرحله چهار (پیاده‌نظام) جواب نهایی مسئله.

t: ضریب خستگی.

α: اندازه گام‌ها در حرکت به سمت فرمانده در مرحله دو.

β: اندازه گام‌ها در حرکت به سمت فرمانده در مرحله سه/سواره‌نظام (حرکت با اندازه گام‌های بزرگ).

γ : اندازه گام‌ها در حرکت به سمت فرمانده در مرحله چهار/پیاده‌نظام (حرکت با اندازه گام‌های کوچک).

ویژگی‌های این الگوریتم را نیز می‌توان به‌صورت زیر جمع‌بندی کرد:

- *در مرحله یک، با تعداد زیاد جواب‌های تصادفی فضای جواب بمباران و هم‌چنین در مراحل بعدی نیز از جواب‌های تصادفی برای ایجاد جمعیت استفاده می‌شود؛ بنابراین کل فضای جواب بارها درنوردیده می‌شود. این موضوع باعث افزایش احتمال عبور از بهینه‌های محلی می‌شود.*
- *منطقه (فضای جواب) به ۴ منطقه به‌صورت مساوی شده و هر یك از این مناطق به یک گروه برای جستجو تخصیص داده می‌شود. این موضوع احتمال دستیابی به جواب بهینه/نزدیک بهینه سراسری و عبور از بهینه‌های محلی را بالا می‌برد.*
- *حرکت اعضای جمعیت سواره با اندازه گام‌های بزرگ جهت ایجاد تنوع / استخراج[1] و حرکت اعضای جمعیت پیاده با اندازه گام‌های کوچک برای بهره‌برداری/کراس آور[2] الگوریتم را برای یافتن جواب بهینه یاری می‌کنند.*
- *استفاده از بهترین جواب‌های هر مرحله در مراحل بعدی به‌عنوان یکی از جواب‌های اولیه کارایی الگوریتم را بالا برده است.*

[1] Diversification
[2] Intensification

– این الگوریتم از سرعت بالاتری نسبت به سایر الگوریتم‌ها (به جزء WOA) برخوردار است، به‌طوری‌که در همه ۲۳ تست انجام‌شده، ازنظر زمان اجرا (سرعت) رتبه دوم را (در میان نه الگوریتم تحت بررسی) به دست آورده است.

ویژگی‌های فوق امکان عملکرد بهتر الگوریتم پیشنهادی را فراهم آورده است. تست‌های انجام‌شده نیز این امر را تائید می‌کند؛ به‌طوری‌که در تست‌ها و رتبه‌بندی اولیه انجام‌شده، *MOA* در مقایسه با سایر الگوریتم‌ها، ازنظر عملکردی در سطح بالاتری قرارگرفته است.

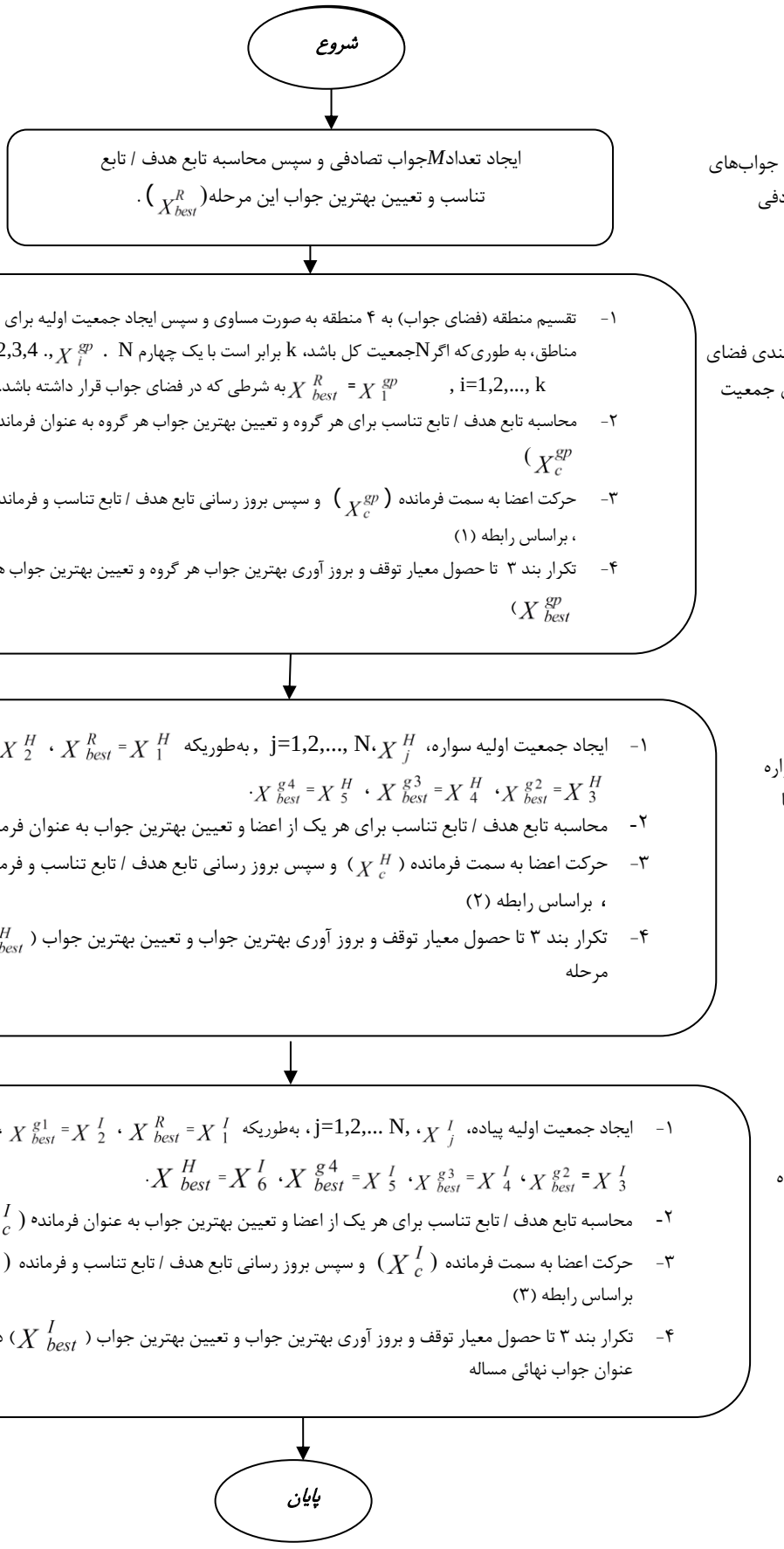
شکل ۲ فلوچارت الگوریتم پیشنهادی را نشان می‌دهد.

۳- روش‌شناسی تحقیق

در این مقاله یک الگوریتم بهینه‌سازی جدید ارائه‌شده است. عملکرد این الگوریتم پیشنهادی، با بهره‌گیری از ۲۳ تابع نمونه استاندارد به شرح **جدول ۲** در مقایسه با هشت الگوریتم فرا ابتکاری شامل: *ABC*، *PSO*، *ICA*، *SFLA*، *GA*، *GWO*، *WOA* و *GOA* مورد ارزیابی قرار گرفت. برای این منظور هر یک از توابع به‌وسیله الگوریتم پیشنهادی و الگوریتم‌های محک با نرم‌افزار متلب (*MATLAB*) حل شد. مقایسه در دو شاخص میانگین تابع هدف (دقت) و میانگین زمان محاسباتی (سرعت) صورت گرفته و رتبه‌بندی بر این اساس انجام شد. سپس برای بررسی‌های آماری و مقایسه میانگین‌ها از روش تحلیل واریانس و نرم‌افزار *SPSS* و برای تنظیم پارامترها به‌وسیله روش تاگوچی از نرم‌افزار مینی‌تب (*MINITAB*) استفاده شد.

در **جدول ۲** تعداد ۲۳ تابع نمونه (تست) با ذکر مشخصات آن‌ها از قبیل نام تابع، دامنه، تعداد متغیرهای تابع، تک نمایی یا چند نمایی بودن تابع، فرمول و مقدار بهینه سراسری هر تابع، فهرست شده‌اند (کاوه و طلاطاهری[1]، ۲۰۱۰؛ میر جلیلی و همکاران، ۲۰۱۴؛ یان و هاوو[2]، ۲۰۱۳؛ طبری و احمد، ۲۰۱۷؛ مولگا و اسموتنیکی[3]، ۲۰۰۵). این توابع برای آزمون الگوریتم پیشنهادی در مقایسه با الگوریتم های محک، مورداستفاده قرارگرفته است.

[1] Kaveh and Talatahari
[2] Yan and Hao
[3] Molga and Smutnicki



شکل ۲- فلوچارت الگوریتم بهینه‌سازی نظامی (MOA).

Figure 2- Military Optimization Algorithm (MOA) flowchart.

جدول ۲- توابع نمونه استاندارد به همراه مشخصات آنان.

Table 2- Standard test functions with their specifications.

تابع	دامنه	بعد	ویژگی	فرمول تابع		بهینه سراسری
F1 (Sphere)	[-100 , 100]	30	تک نمایی	$\sum_{i=1}^{n} X_i^2$	F(x) =	F(x) = 0
F2 (Sum of different power)	[-1 , 1]	30	تک نمایی	$\sum_{i=1}^{n} \lvert X_i \rvert^{i+1}$	F(x) =	F(x) = 0
F3 Six-hump camel) (back	[-3 , 3]	2	چند نمایی	$(4-2.1X_1^2+X_1^{4/3})\cdot X_1^2+X_1X_2+(-4+4X_2^2)\cdot X_2^2$	F(x) =	F(x) =-1.0316
F4 (Drop wave)	[-5.12 , 5.12]	2	چند نمایی	$\frac{1+\cos(12\sqrt{X_1^2+X_2^2})}{0.5(X_1^2+X_2^2)+2}$	F(x) =	F(x) = -1
F5 weighted sphere) (model	[-5.12 , 5.12]	30	تک نمایی	$\sum_{i=1}^{n} i\cdot X_i^2$	F(x) =	F(x) = 0
F6 (Goldstein-Price)	[-2, 2]	2	چند نمایی	$(1+(X_1+X_2+1)^2\cdot(19-14X_1+3X_1^2-14X_2+6X_1X_2+3X_2^2))\cdot(30+(2X_1-3X_2)^2\cdot(18-32X_1+12X_1^2+48X_2-36X_1X_2+27X_2^2))$	F(x) =	F(x) = 3
F7 (Griewangk)	[-600 , 600]	30	چند نمایی	$\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	F(x) =	F(x) = 0
F8 Moved axis) parallel hyper- (ellipsoid	[-5.12 , 5.12]	30	تک نمایی	$\sum_{i=1}^{n} 5*i\cdot X_i^2$	F(x) =	F(x) = 0
F9 (Rastrigin)	[-5.12 , 5.12]	30	چند نمایی	$10\cdot n+\sum_{i=1}^{n}(x_i^2+10\cos(2\cdot\pi\cdot x_i))$	F(x) =	F(x) = 0
F10 (Rosenbrock)	[-2.048 , 2.048]	30	تک نمایی	$\sum_{i=1}^{n} 100(x_{i+1}-x_i^2)^2+(1-x_i)^2$	F(x) =	F(x) = 0
F11 (Aluffi-Pentiny)	[-10 , 10]	2	تک نمایی	$1/4X_1^4-1/2X_1^2+1/10X_1+1/2X_2^2$	F(x) =	-F(x) = 0.352386
F12 (Becker and Lago)	[-10 , 10]	2	چند نمایی	$(\lvert x_1\rvert-5)^2+(\lvert x_2\rvert-5)^2$	F(x) =	F(x) =0
F13 (Bohachevsky1)	[-100 , 100]	2	چند نمایی	$X_1^2+2X_2^2-0.3\cos(3\pi X_1)-0.4\cos(4\pi X_2)+0.7$	F(x) =	F(x) = 0
F14 (Bohachevsky2)	[-50 , 50]	2	چند نمایی	$X_1^2+2X_2^2-0.3\cos(3\pi X_1)(4\pi X_2)+0.3$	F(x) =	F(x) = 0
F15 (Booth)	[-10 , 10]	2	چند نمایی	$(X_1+2X_2-7)^2+(2X_1+X_2-5)^2$	F(x) =	F(x) = 0
F16 (Branin)	[-5 , 15]	2	چند نمایی	$(X_2-\frac{5.1}{4\pi}X_1^2+\frac{5}{\pi}X_1-6_1)^2+10(1-\frac{1}{8\pi})\cos X_1+10$	F(x) =	F(x) = 0.39789
F17 (Cosine mixture)	[-1 , 1]	4	چند نمایی	$\sum_{i=1}^{n} X_i^2-0.1\sum_{i=1}^{n}\cos(5\pi X_i)$	F(x) =	F(x) = -0.4
F18 (Exponential)	[-1 , 1]	30	تک نمایی	$-\exp(-0.5\sum_{i=1}^{n} X_i^2)$	F(x) =	F(x) =0
F19 (Quartic)	[-1.28 , 1.28]	30	تک نمایی	$\sum_{i=1}^{n}(ix_i)^4+Random(0,1)$	F(x) =	F(x) = 0

جدول ۲- (ادامه).

Table 2- (Continued).

تابع	دامنه	بعد	ویژگی	فرمول تابع		بهینه سراسری
F20 (Step)	[-100 , 100]	30	تک نمایی	$\sum_{i=1}^{n}(\lvert X_i+0.5\rvert)^2$	F(x) =	F(x) =0
F21 (Beale)	[-4.5 , 4.5]	2	تک نمایی	$(1.5-X_1+X_1X_2)^2$ $+(2.25-X_1+X_1X_2^2)^2$ $+(2.625-X_1+X_1X_2^3)^2$	F(x) =	F(x) = 0
F22 (Colville)	[-10 , 10]	4	تک نمایی	$100(X_1^2+X_2)^2+(X_1-1)^2+$ $(X_3-1)^2+90(X_3^2+X_4)^2+$ $10.1(X_2-1)^2+X_4-1)^2+$ $19.8(X_2-1)(X_4-1)$	F(x) =	F(x) = 0
F23 (Matyas)	[-10 , 10]	2	تک نمایی	$0.26(X_1^2+X_2^2)-0.48X_1X_2$	F(x) =	F(x) = 0

۴- اجرا، تحلیل و یافته‌های تحقیق

در این بخش توابع تست که در بخش سوم نشان داده شد، پس از تنظیم پارامترها، توسط الگوریتم پیشنهادی و الگوریتم‌های محک حل و نتایج مورد تحلیل قرارگرفته و رتبه‌بندی انجام می‌گیرد.

۴-۱- تنظیم پارامترها

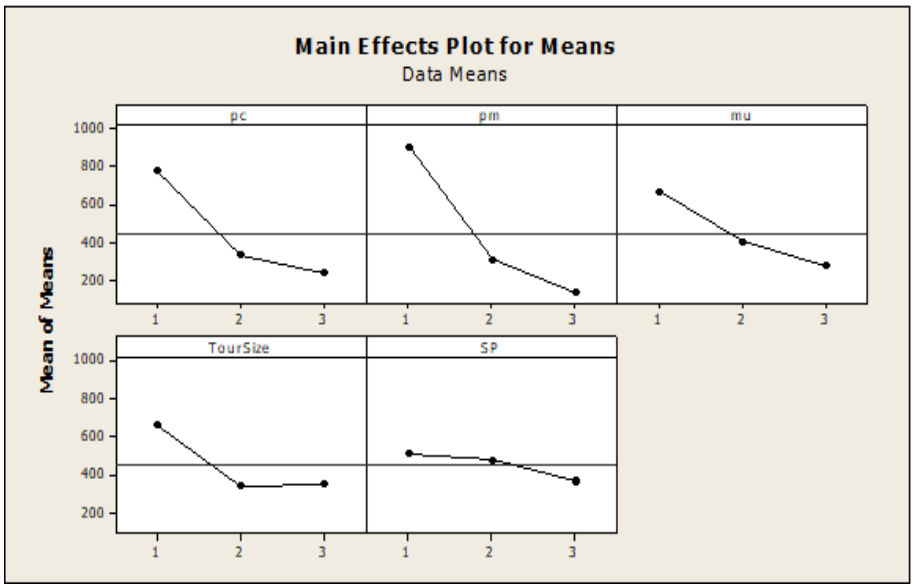
خروجی الگوریتم‌های فرا ابتکاری تحت تأثیر پارامترهای ورودی آن‌هاست. بنابراین در این مقاله، برای تنظیم پارامترها از روش تاگوچی استفاده‌شده است. به‌عنوان نمونه تنظیم پارامترهای الگوریتم ژنتیک، مطابق **جدول ۳** در سه سطح برای پارامترهای *pc*، *pm*، *mu*، *TourSize*، *SP* انجام‌شده است.

جدول ۳- تنظیم پارامترهای الگوریتم ژنتیک.

Table 3- Adjustment of genetic algorithm parameters.

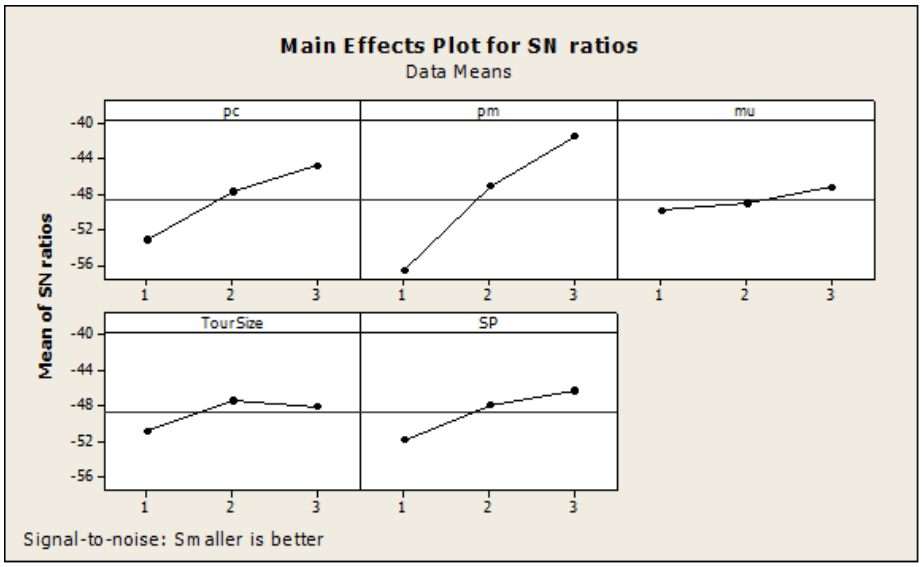
ردیف	پارامتر	شرح	سطح اول	سطح دوم	سطح سوم
1	pc	Crossover Percentage	0.7	0.8	0.9
2	pm	Mutation Percentage	0.1	0.2	0.3
3	mu	Mutation Rate	0.01	0.02	0.03
4	TourSize	TournamentSize	2	3	4
5	SP	Selection Pressure	5	8	10

شکل‌های ۳ و **۴** آنالیز نتایج به‌دست‌آمده از اجرای آزمایش تاگوچی توسط نرم‌افزار *Minitab* است که با توجه به آن‌ها فاکتورهای ۱، ۲، ۳ و ۵ در سطح سوم و فاکتور ۴ در سطح دوم، بهترین عملکرد الگوریتم را ارائه داده‌اند.



شکل ۳- نمودار میانگین میانگین‌ها برای هر سطح از پارامترهای الگوریتم ژنتیک.

Figure 3- Average diagram of means for each level of genetic algorithm parameters.



شکل ۴- نمودار میانگین S/N برای هر سطح از پارامترهای الگوریتم ژنتیک.

Figure 4- Graph of the average S / N for each level of genetic algorithm parameters.

۲-۴- اجرای الگوریتم‌ها برای حل توابع

در این بخش توابع تست (که در جدول ۲ نشان داده شد) به‌وسیله الگوریتم پیشنهادی (MOA) و الگوریتم‌های ABC، PSO، ICA، SFLA، GA، GWO، WOA و GOA که همگی از الگوریتم‌های مبتنی بر جمعیتی از جواب‌ها[1] هستند، پس از تنظیم پارامترها، با تعداد جمعیت و تکرارهای برابر، به‌وسیله نرم‌افزار متلب (MATLAB) با ۳۰ بار اجرا (Run) حل شد. تحقیقات پیشین نیز بر اساس همین تعداد تکرار (۳۰ بار اجرا) به مقایسه الگوریتم‌ها پرداخته‌اند. در این خصوص می‌توان به این پژوهش‌ها اشاره کرد: کلونی زنبور مصنوعی (کارابوگا و باستورک، ۲۰۰۷)، بهینه‌ساز گرگ خاکستری (میر جلیلی و همکاران، ۲۰۱۴)، جستجوی سیستم شارژ شده (کاوه و طلاطاهری، ۲۰۱۰) و حرکت یون‌ها (جاویدی و همکاران، ۲۰۱۵). ازاین‌رو، ما نیز در مطالعه حاضر به‌تبع تحقیقات قبلی از ۳۰ بار اجرا جهت ارزیابی و مقایسه میان الگوریتم‌های موردمطالعه استفاده کردیم.

[1] Population Based

نتایج محاسبات ازنظر دو شاخص میانگین جواب‌ها (دقت) و زمان اجرای الگوریتم (سرعت) و رتبه‌بندی، در **جدول ۴** نشان داده‌شده است.

جدول ۴- نتایج اجرای الگوریتم‌ها برای حل توابع تست.

Table 4- Results of implementation of algorithms for solving test functions.

توابع			الگوریتم							
			ABC	PSO	SFLA	ICA	GA	GWO	WOA	GOA	MOA
Sphere	n=30	میانگین جواب‌ها	1.526	1.823E-10	9.098E-07	3.833E-02	.215	5.198E-63	5.081E-02	1.196	0
		رتبه	8	3	4	5	9	2	6	7	1
		میانگین زمان اجرا	7 863	4.461	14.4 8	4.048	8.075	3.610	0.757	12.636	2.609
		رتبه	6	5	9	4	7	3	1	8	2
Sum of different power	n=30	میانگین جواب‌ها	.739E-07	1.437E-39	1.970E-18	3.975E-16	2.920E-09	2.718E-186	26.036	4.947E-08	0
		رتبه	8	3	4	5	6	2	9	7	1
		میانگین زمان اجرا	7.610	4.86	18.465	4.678	5.659	4.054	1.768	13.091	3.346
		رتبه	7	5	9	4	6	3	1	8	2
camel	n=2	میانگین جواب‌ها	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
		رتبه	1	1	1	1	1	1	1	1	1
Six-hump back		میانگین زمان اجرا	4.042	2.931	5.660	2.705	4.003	2.412	0.575	10.300	1.938
		رتبه	7	5	8	4	6	3	1	9	2
Dropwave	n=2	میانگین جواب‌ها	-0.99979	-1	-1	-1	-1	-0.99363	-0.97235	-0.97663	-1
		رتبه	2	1	1	1	1	3	5	4	1
		میانگین زمان اجرا	4.274	2.996	5.698	2.589	4.416	2.296	0.556	9.385	1.858
		رتبه	6	5	8	4	7	3	1	9	2
hyper-parallel ellipsoid	n=30	میانگین جواب‌	8.563E-02	4.925E-12	4.476E-08	1.236E-03	?E-01	3.318E-64	0	5. 26	0
		رتبه	6	3	4	5	7	2	1	8	1
		میانگین زمان اجرا	6.268	4.647	15.832	4.062	6.922	3.605	0.911	14.403	2.798
		رتبه	6	5	9	4	7	3	1	8	2

3	3	3	3.00002		3	3	3	3	میانگین جواب‌ها	n=2	Goldstein-Price's
1	1	1	2	1	1	1	1	1	رتبه		
1.827	9.373	0.558	2.346	3.761	2.555	5.514	3.099	2	میانگین زمان اجر		
2	9	1	3	6	4	8	5	7	رتبه		
0	7.054E-01	0	1.798E-03	1.09444	1.123E-01	9.437E-03	1.660E-02	00885	میانگین جواد	n=30	Griewangk
1	6	1	2	8	5	3	4	7	رتبه		
2.939	13.502	1.073	3.628	4.405	4.851	8.344	5.136	962	میانگین زمان اجر		
2	9	1	3	6	4	8	5	7	رتبه		
0	16. 695	0	2.757E-63	1.468E-01	8.990E-03	3.120E-07	1.074E-11	3.371E-01	میانگین جواد	n=30	Moved axid parallel hyper
1	8	1	2	7	5	4	3	6	رتبه		
2.887	12.835	1.159	3.587	7.296	4.291	18.324	4.907	32	میانگین زمان اجر		
2	8	1	3	6	4	9	5	7	رتبه		
0	81.659	0	7.070		38.743	58.769	35.09742	213.697	میانگین جواد	n=30	Rastrigin
1	7	1	3	2	5	6	4	8	رتبه		
2.723	13.081	1.129	3.719	7.422	4.168	18.170	4.968		میانگین زمان اجر		
2	8	1	3	7	4	9	5	6	رتبه		
1.216E-31	9.044 -12	2. 85E+01	9.8005E-07	0E-03	1.815E-03	2.571E-06	1.117E-13	945E-07	میانگین جواد	n=30	Rosenbrock
1	3	9	5	8	7	6	2	4	رتبه		
2.280	12.612	1.313	2.869	4.655	3.965	8.372	3.922	7.575	میانگین زمان اجر		
2	9	1	3	6	5	8	4	7	رتبه		
-0.35239	-0.35239	-0.35239	-0.35239	-0.35239	-0.35239	-0.35239	-0.35239	-0.35239	میانگین جواب‌ها	n=2	AluffiPentiny
1	1	1	1	1	1	1	1	1	رتبه		
1.833	9.315	0.571	2.554	3.798	2.758	6.124	2.982	4.351	میانگین زمان اجر		
2	9	1	3	6	4	8	5	7	رتبه		
0	6.592E-04	0	1.800E-06		2.518E-23	0	0	1.297E-08	میانگین جواب‌ها	n=2	Beck eran

1	5	1	4	1	2	1	1	3	رتبه		
1.781	9.308	0.523	2.548	3.757	2.956	5.572	3.177	5.729	میانگین زمان اجرا		
2	9	1	3	6	4	7	5	8	رتبه		
0	7.063E-07	0	0	0	0	0	0	4.448E-06	میانگین جواب‌ها	n=2	Bohachevsky1
1	2	1	1	1	1	1	1	3	رتبه		
1.713	10.110	0.551	2.444	3.696	2.642	5.564	2.971	5.855	میانگین زمان اجرا		
2	9	1	3	6	4	7	5	8	رتبه		
0	2.272E-07	5.094E-02	0	0	0	0	0	3.258E-07	میانگین جواب‌	n=2	Bohachevsky2
1	2	4	1	1	1	1	1	3	رتبه		
1.723	10.090	0.532	2.471	3.679	2.715	5.586	3.138	4.407	میانگین زمان اجرا		
2	9	1	3	6	4	8	5	7	رتبه		
0	1.367E-09	6.300E-04	6.624E-07	0	6.537E-15	0	0	1.764E-09	میانگین جواب‌ها	n=2	Booth
1	3	6	5	1	2	1	1	4	رتبه		
1.862	9.900	0.551	2.545	3.528	2.671	5.412	2.979	5.808	میانگین زمان اجرا		
2	9	1	3	6	4	7	5	8	رتبه		
0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	39789	میانگین جواب‌ها	n=2	Branins
1	1	1	1	1	1	1	1	1	رتبه		
1.658	9.519	0.546	2.854	3.636	2.749	5.639	3.118	4.405	میانگین زمان اجرا		
2	9	1	4	6	3	8	5	7	رتبه		
-0.4	-0.3261	-0.4	-0.4	-0.4	-0.4	-0.4	-0.4	-0.4	میانگین جواب‌ها	n=2	Cosinemixture
1	2	1	1	1	1	1	1	1	رتبه		
2.701	12.215	1.111	2.842	4.895	3.759	7.127	4.584	7.927	میانگین زمان اجرا		
2	9	1	3	6	4	7	5	8	رتبه		
-1	-0.99999	-1	-1	-0.99957	-1	-1	-1	-0.99992	میانگین جواب‌ها	n=30	Exponential
1	2	1	1	4	1	1	1	3	رتبه		
2.462	13.128	1.085	3.476	5.254	3.699	8.300	4.542	5.873	میانگین زمان اجرا		
2	9	1	3	6	4	8	5	7	رتبه		

6.009E-04	1.344E-01	2.977E-03	2.076E-03	.986E-03	1.276E-01	2.757E-02	1.672E-02	'.804E-02	میانگین جواب‌ها		
1	9	3	2	4	8	6	5	7	رتبه	n=30	Quartic
2.780	13.489	1.567	3.432	5.159	4.063	10.148	4.852	6.379	میانگین زمان اجرا		
2	9	1	3	6	4	8	5	7	رتبه		
1.170E-10	1.336E+00	2.483E-03	7.659E-01	3.331E+00	5.885E-02	1.375E-06	2.205E-10	.710E+00	میانگین جواب‌ها		
1	7	4	6	9	5	3	2	8	رتبه	n=30	Step
2.671	13.266	1.584	3.452	6.999	3.818	13.850	4.646	7.710	میانگین زمان اجرا		
2	8	1	3	6	4	9	5	7	رتبه		
0	5.431E-03	0	2.164E-07	2.903E-05	2.926E-11	0	8.570E-28	.345E-07	میانگین جواب‌ها		
1	7	1	5	6	3	1	2	4	رتبه	n=2	Beale
1.810	9.299	0.577	2.672	4.038	3.898	7.033	3.104	5.636	میانگین زمان اجرا		
2	9	1	3	6	5	8	4	7	رتبه		
2.136E-10	1.182E-04	1.723E+00	1.616E+00	1.742E-01	4.842E-02	4.481E-04	2.722E-02	1.660E-01	میانگین جواب‌ها		
1	2	9	8	7	5	3	4	6	رتبه	n=2	Colville
2.373	12.348	0.715	3.344	4.842	4.996	9.554	3.820	4.950	میانگین زمان اجرا		
2	9	1	3	5	7	8	4	6	رتبه		
0	6.089E-11	0	5.649E-192	5.745E-22	1.999E-14	2.366E-57	8.505E-30	1.628E-10	میانگین جواب‌ها		
1	7	1	2	5	6	3	4	8	رتبه	n=2	Matyas
1.796	9.768	0.534	2.623	3.840	3.620	7.488	3.127	4.261	میانگین زمان اجرا		
2	9	1	3	6	5	8	4	7	رتبه		

جدول ۴ خروجی ۳۰ بار اجرای (*Run*) الگوریتم‌ها برای حل توابع تست بوده و لذا پاسخ هر الگوریتم برای هر یک از توابع را در دو شاخص میانگین ۳۰ بار اجرا و میانگین زمان اجرای الگوریتم در ۳۰ بار اجرا را نشان می‌دهد. علاوه بر این، رتبه هر الگوریتم در هر تابع نیز در **جدول ۴** قابل‌مشاهده است. لذا با توجه به تعداد توابع درمجموع در هر شاخص ۲۳ تست انجام شد و عملکرد و رتبه هر الگوریتم در ۲۳ تست در جدول فوق به‌صورت جداگانه نشان داده‌شده است.

مقایسه آماری نتایج به‌صورت زیر است:

- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۸ تابع عملکرد بهتر از الگوریتم ABC و در ۵ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، MOA (الگوریتم پیشنهادی) در همه ۲۳ تابع دارای عملکرد بهتری نسبت به الگوریتم ABC بوده است.*
- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۲ تابع عملکرد بهتری از الگوریتم PSO و در ۱۱ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم در همه ۲۳ تابع، شاهد عملکرد بهتر الگوریتم MOA نسبت به الگوریتم PSO بوده‌ایم.*
- *بر اساس نتایج، الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۱ تابع عملکرد بهتر از الگوریتم SFLA و در ۱۲ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، MOA در همه ۲۳ تابع بهتر از الگوریتم SFLA عمل کرده است.*
- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۴ تابع عملکرد بهتر از الگوریتم ICA و در ۹ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، MOA (الگوریتم پیشنهادی) در همه ۲۳ تابع دارای عملکرد بهتری نسبت به الگوریتم ICA بوده است.*
- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۳ تابع عملکرد بهتر از الگوریتم GA و در ۱۰ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، در همه ۲۳ تابع شاهد عملکرد بهتر الگوریتم MOA نسبت به الگوریتم GA بوده‌ایم.*
- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۶ تابع عملکرد بهتر از الگوریتم GWO و در ۷ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، در همه ۲۳ تابع شاهد عملکرد بهتر الگوریتم MOA نسبت به الگوریتم GWO بوده‌ایم.*
- *الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۹ تابع عملکرد بهتر از الگوریتم WOA و در ۱۴ تابع عملکرد برابر آن داشته است. لیکن در شاخص سرعت اجرای الگوریتم، WOA در همه ۲۳ تابع دارای عملکرد بهتری نسبت به MOA (الگوریتم پیشنهادی) بوده است.*
- *بر اساس نتایج، الگوریتم پیشنهادی در شاخص یافتن جواب بهینه سراسری و یا نزدیک شدن آن، در ۱۹ تابع عملکرد بهتر از الگوریتم GOA و در ۴ تابع عملکرد برابر آن داشته است. همچنین در شاخص سرعت اجرای الگوریتم، MOA در همه ۲۳ تابع بهتر از الگوریتم GOA عمل کرده است.*

در ادامه نتایج با استفاده از روش تحلیل واریانس موردبررسی قرارگرفته و میانگین‌ها مورد مقایسه قرار می‌گیرند.

۳-۴- مقایسه نتایج با استفاده از تحلیل واریانس

در این بخش، نتایج نشان داده در **جدول ٤** با استفاده از روش تحلیل واریانس موردبررسی قرار می‌گیرد. **جدول ۵** خلاصه تحلیل‌های انجام‌شده را نشان می‌دهد.

جدول ۵- خلاصه تحلیل واریانس.
Table 5- Summary of analysis of variance.

توابع	الگوریتم		ABC	PSO	SFLA	ICA	GA	GWO	WOA	GOA
Sphere	MOA	میانگین جواب‌ها	*				*			*
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Sum of different power	MOA	میانگین جواب‌ها							*	
		میانگین زمان اجرا	*	*	*	*	*		*	*
Six-hump camel back	MOA	میانگین جواب‌ها								
		میانگین زمان اجرا	*	*	*	*	*		*	*
Dropwave	MOA	میانگین جواب‌ها								*
		میانگین زمان اجرا	*	*	*	*	*		*	*

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ۶، شماره ۳، پاییز ۱۴۰۰، صفحه: ۳۲۹-۳۰۴

جدول ۵- (ادامه).

Table 5- (Continued).

توابع	الگوریتم		ABC	PSO	SFLA	ICA	GA	GWO	WOA	GOA
Parallel Hyper-Ellipsoid	MOA	میانگین جواب‌ها	*	*	*	*	*			*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Goldstein-Price's	MOA	میانگین جواب‌ها						*		
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Griewangk	MOA	میانگین جواب‌ها	*			*	*			*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Moved axid parallel hyper	MOA	میانگین جواب‌ها								*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Rastrigin	MOA	میانگین جواب‌ها	*	*	*	*				*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Rosenbrock	MOA	میانگین جواب‌ها							*	
		میانگین زمان اجرا	*	*	*	*	*		*	*
AluffiPentiny	MOA	میانگین جواب‌ها								
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
BeckerandLago	MOA	میانگین جواب‌ها								*
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Bohachevsky1	MOA	میانگین جواب‌ها	*							
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Bohachevsky2	MOA	میانگین جواب‌ها							*	
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Booth	MOA	میانگین جواب‌ها							*	
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Branins	MOA	میانگین جواب‌ها								
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Cosinemixture	MOA	میانگین جواب‌ها								*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Exponential	MOA	میانگین جواب‌ها	*				*			
		میانگین زمان اجرا	*	*	*	*	*		*	*
Quartic	MOA	میانگین جواب‌ها	*		*	*				*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Step	MOA	میانگین جواب‌ها	*				*	*		*
		میانگین زمان اجرا	*	*	*	*	*		*	*
Beale	MOA	میانگین جواب‌ها								*
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Colville	MOA	میانگین جواب‌ها						*	*	
		میانگین زمان اجرا	*	*	*	*	*	*	*	*
Matyas	MOA	میانگین جواب‌ها	*							
		میانگین زمان اجرا	*	*	*	*	*	*	*	*

در **جدول ۵** علامت ستاره (*) نشان‌دهنده معنادار بودن اختلاف بین میانگین‌ها است. به‌عنوان‌مثال: در خصوص تابع اسفایر (*Sphere*)، شاهد اختلاف معنادار میانگین مقدار تابع هدف (جواب‌ها) *MOA* با مقادیر مربوط به الگوریتم‌های *ABC*، *GA* و *GOA* بوده‌ایم؛ این در حالی است که میانگین محاسبه‌شده با مقادیر مربوط به سایر الگوریتم‌ها اختلاف معناداری نداشته است. به‌صورت مشابه، اختلاف میانگین زمان اجرا بین *MOA* با همه الگوریتم‌های موردمطالعه معنادار بوده است؛ بنابراین می‌توان نتیجه‌گیری کرد که عملکرد الگوریتم پیشنهادی ازنظر شاخص میانگین جواب‌ها در تابع اسفایر بهتر از *ABC*، *GA* و *GOA* و برابر با سایر الگوریتم‌ها است. ازنظر میانگین زمان اجرای الگوریتم نیز عملکرد الگوریتم پیشنهادی بهتر از سایر الگوریتم‌ها به جزء *WOA* می‌باشد (به **جدول ۴** رجوع شود). این تحلیل را می‌توان به سایر توابع نیز تعمیم داد.

جدول ۶ خلاصه مقایسات را به‌صورت آماری نشان می‌دهد.

جدول ۶- خلاصه آماری مقایسات.

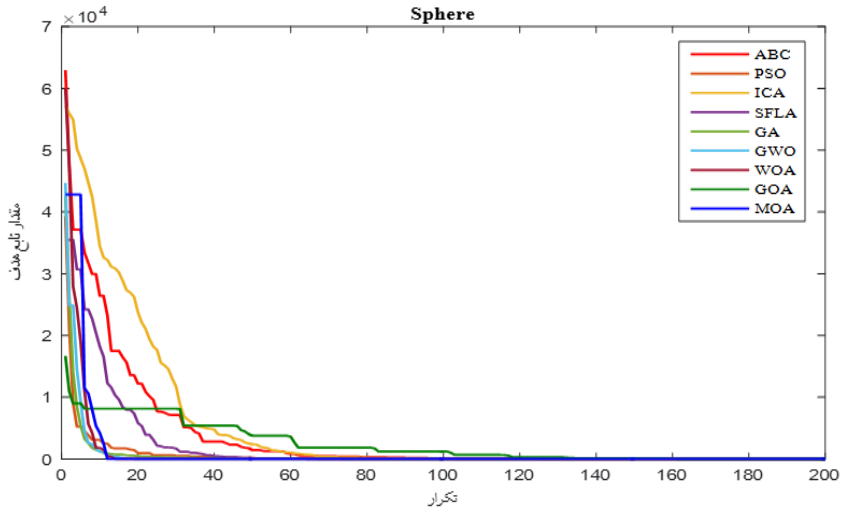
Table 6- Statistical summary of comparisons.

	رتبه / تحلیل واریانس	شاخص‌ها	بهتر از سایر الگوریتم‌ها	برابر با سایر الگوریتم‌ها	بدتر از سایر الگوریتم‌ها	جمع
MOA	بر اساس رتبه	میانگین جواب‌ها	112	72	0	184
		میانگین زمان اجرا	161	0	23	184
	بر اساس تحلیل واریانس	میانگین جواب‌ها	37	147	0	184
		میانگین زمان اجرا	149	12	23	184

جدول ۷ سرعت/زمان همگرایی الگوریتم‌های تحت بررسی را نشان می‌دهد.

مطابق با **جدول ۷** الگوریتم پیشنهادی ازنظر شاخص سرعت همگرایی در ۱۰ تست انجام‌شده به‌وسیله توابع مختلف در مقایسه با سایر الگوریتم‌ها، ٤ بار رتبه اول، ٤ بار رتبه دوم و ۲ بار نیز رتبه چهارم را کسب نموده است؛ بنابراین می‌توان نتیجه گرفت که *MOA* در این شاخص نیز عملکرد مطلوب و قابل قبولی داشته است.

شکل‌های ۵ تا ۸ حرکت به سمت جواب بهینه توسط الگوریتم‌ها در طی تکرارها را در توابع: *Sphere*، *Griewangk*، *Rastrigin* و *Step* نشان می‌دهد.



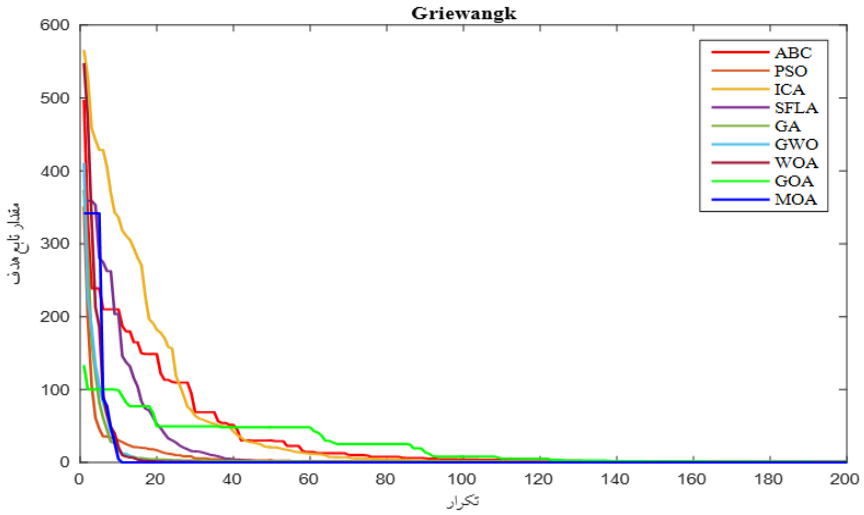
شکل ۵- حرکت به سمت جواب بهینه توسط الگوریتم‌ها در تابع Sphere.

Figure 5- The movement towards optimal answer by the algorithms in solving Sphere function.

جدول ۷- زمان همگرایی الگوریتم‌ها.

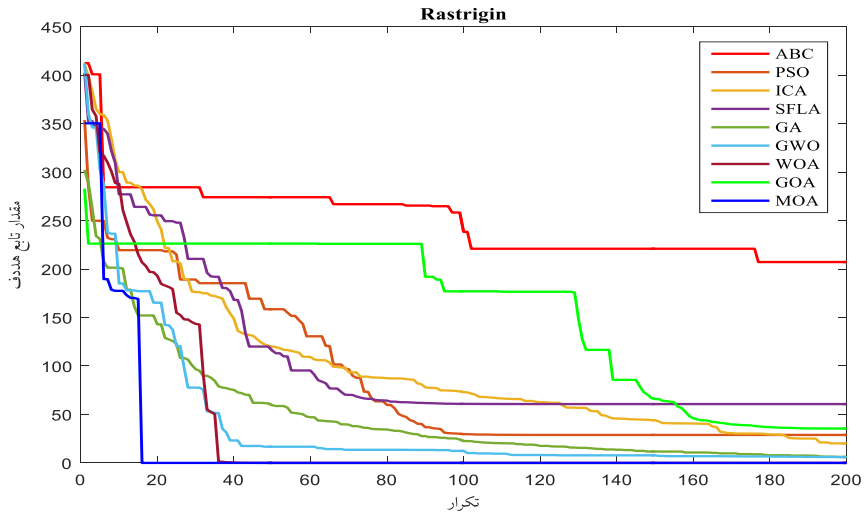
Table 7- Convergence time of algorithms.

توابع			الگوریتم								
			ABC	PSO	SFLA	ICA	GA	GWO	WOA	GOA	MOA
Six-hump camel back	n=2	میانگین زمان	0.44942	.12127	0.28082	0.19204	0.34752	0.23552	0.04779	5.75475	0.25835
		رتبه	7	8	5	2	6	3	1	9	4
Dropwave	n=2	میانگین زمان همگرایی	1.7418404	.176813	0.78288501	1.9429044	0.843803	0.352883	0.077209	8.444148	0.071106
		رتبه	6	7	4	8	5	3	2	9	1
Goldstein-Price’s	n=2	میانگین زمان همگر	2.02081	.14196	0.30992	0.42014	.45777	17	0.08582	7.64782	0.06484
		رتبه	8	6	3	4	5	7	2	9	1
AluffiPentiny	n=2	میانگین زمان همگ	0.99269	.96059	0.33647	0.29537	0.49474	0.66168	0.03973	9.29376	0.24223
		رتبه	8	7	4	3	5	6	1	9	2
BeckerandLago	n=2	میانگین زمان هم	54953	.86262	1.45543	6.64886	.02186	2.51246	0.04170	9.91067	0.11525
		رتبه	6	7	3	8	4	5	1	9	2
Bohachevsky1	n=2	میانگین زمان ه	4.30323	.08784	0.90778	2.07695	1.25824	0.33384	0.06818	9.06846	0.09943
		رتبه	8	7	4	6	5	3	1	9	2
Bohachevsky2	n=2	میانگین زمان همگرایی	82400	.29976	0.99281	1.99961	1.17325	0.34825	.10734	9.70148	0.09878
		رتبه	8	7	4	6	5	3	2	9	1
Branins	n=2	میانگین زمان همگرایی	1.35929	.97168	26539	.36101	.41598	2.67970	0.54659	18543	0.38606
		رتبه	8	7	1	3	5	2	6	9	4
Cosinemixture	n=2	میانگین زمان همگ	1.71051	.02262	0.29508	0.84724	0.68365	0.14043	0.10828	.1.09122	0.11376
		رتبه	8	7	4	6	5	3	1	9	2
Exponential	n=30	میانگین زمان همگ	3.66354	.51886	5.91946	5520	4.36371	0.66624	15839	13.44436	0.12804
		رتبه	6	5	8	4	7	3	2	9	1



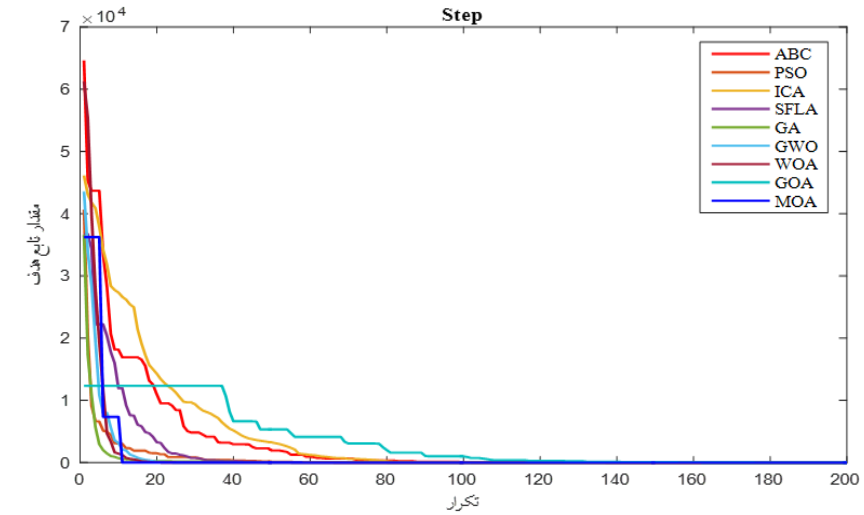
شکل ۶- حرکت به سمت جواب بهینه توسط الگوریتم‌ها در تابع Griewangk .

Figure 6- The movement towards optimal answer by the algorithms in solving Griewangk function.



شکل ۷- حرکت به سمت جواب بهینه توسط الگوریتم‌ها در تابع Rastrigin.

Figure 7- The movement towards optimal answer by the algorithms in solving Rastrigin function.



شکل ۸- حرکت به سمت جواب بهینه توسط الگوریتم‌ها در تابع Step .

Figure 8- The movement towards optimal answer by the algorithms in solving Step function.

۵- نتیجه‌گیری

در این مقاله به معرفی و آزمون یك الگوریتم فرا ابتکاری (با عملکرد بهتر/قابل‌رقابت با سایر الگوریتم‌ها)، با عنوان الگوریتم بهینه‌سازی نظامی (*MOA*) پرداخته‌ایم. الگوریتم پیشنهادی مبتنی بر جمعیت، جستجوی تصادفی، تقسیم فضای جواب به چند منطقه و تخصیص بخشی از جمعیت به هر منطقه، جستجوی سواره‌نظام و جستجوی پیاده‌نظام استوار است. عملکرد این الگوریتم پیشنهادی به‌وسیله توابع تست استاندارد موردسنجش قرار گرفت و سپس با هشت الگوریتم محک شامل: *ABC*، *PSO*، *ICA*، *SFLA*، *GA*، *GWO*، *WOA* و *GOA*، ازنظر شاخص میانگین جواب‌ها، زمان اجرای الگوریتم و سرعت/زمان همگرایی مقایسه شد. نتایج نشانگر عملکرد مطلوب الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌های موردبررسی است.

مزیت‌های الگوریتم *MOA* به شرح زیر است:

- *سرعت بالای اجرای الگوریتم.*
- *یافتن جواب بهینه سراسری و یا جواب نزدیک بهینه.*
- *عبور از بهینه‌های محلی با احتمال بالا .*
- *قابلیت جستجوی گسترده فضای جواب.*

الگوریتم پیشنهادی با توجه به عملکرد آن در مقایسه با سایر الگوریتم‌ها، می‌تواند در حل برخی از مسائل بهینه‌سازی مورداستفاده قرار گیرد. هم‌چنین، با در نظر گرفتن عملکرد الگوریتم‌ها در هر یک از شاخص‌ها، می‌توان از نتایج این تحقیق در انتخاب الگوریتم برای حل مسائل، مطابق با اولویت و اهمیت هر یک از شاخص‌ها برای تصمیم‌گیرنده، استفاده کرد. نهایتاً استفاده از رویکرد و روش به‌کاررفته در این تحقیق در ارائه الگوریتم بهینه‌سازی جدید، می‌تواند پیشنهادی باشد برای معرفی و ارائه الگوریتم‌های بهینه‌سازی جدیدتر با الهام گیری از سایر پدیده‌های موجود در جهان.

منابع

Abualigah, L., Diabat,O A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, *376*. https://doi.org/10.1016/j.cma.2020.113609

Alam Tabriz, A., Zandieh, M., & Mohammad Rahimi, A. (2008). *Meta-heuristic algorithms in hybrid optimization*. Ishraqi Publications – Saffar.

Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *2007 IEEE congress on evolutionary computation* (pp. 4661-4667). IEEE.

Dehghani, M., Montazeri, Z., Givi, H., Guerrero, J. M., & Dhiman, G. (2020). Darts game optimizer: A new optimization technique based on darts game. *Int. J. Intell. Eng. Syst*, *13*, 286-294.

Dehghani, M., Montazeri, Z., Malik, O. P., Givi, H., & Guerrero, J. M. (2020). Shell game optimization: a novel game-based algorithm. *International journal of intelligent engineering and systems*, *13*(3), 246-255.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, *26*(1), 29-41.

Eusuff, M. M., & Lansey, K. E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of water resources planning and management*, *129*(3), 210-225.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2020). Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft computing*, *19*(1), 1-29.

Ghahramani Nahr, J. (2019). Improving the efficiency and effectiveness of the closed loop supply chain: the Sari Wall optimized algorithm approach and new priority-based encryption. *Journal of decision making and operations research, 4*(4), 299-315. (**In Persian**). http://www.journal-dmor.ir/article_103943.html

Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied soft computing*, *19*, 177-187.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.

Iordache, S. (2010, July). Consultant-guided search: a new metaheuristic for combinatorial optimization problems. *Proceedings of the 12th annual conference on Genetic and evolutionary computation* (pp. 225-232). https://doi.org/10.1145/1830483.1830526

Javidy, B., Hatamlou, A., & Mirjalili, S. (2015). Ions motion algorithm for solving optimization problems. *Applied soft computing*, *32*, 72-79.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, *39*(3), 459-471.

Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in engineering software*, *110*, 69-84.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta mechanica*, *213*(3), 267-289.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942-1948). IEEE.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671-680.

Lam, A. Y., & Li, V. O. (2009). Chemical-reaction-inspired metaheuristic for optimization. *IEEE transactions on evolutionary computation*, *14*(3), 381-399.

Li, X., Zhang, J., & Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural computing and applications*, *24*(7), 1867-1877.

Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, *27*(4), 1053-1073.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

Mohammad Pour Zarandi, M. A. (2013). *Nonlinear optimization*. University of Tehran Press.

Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. Avalable at http://www.sciepub.com/reference/67412

Osaba, E., Diaz, F., & Onieva, E. (2014). Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied intelligence*, *41*(1), 145-166.

Qiao, W., & Yang, Z. (2019). Solving large-scale function optimization problem by using a new metaheuristic algorithm based on quantum dolphin swarm algorithm. *IEEE access*, *7*, 138972-138989.

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design*, *43*(3), 303-315.

Safavi, S. A. A., Pour Jafarian, N., & Safavi, S. A. (2014). *Optimization based on meta-heuristic algorithms*. Academic Publishing Researchers Publications.

Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-based systems*, *75*, 1-18.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization. *International journal of computational science and engineering*, *6*(1-2), 132-140.

Sharifzadeh, H., & Amjady, N. (2014). A review of metaheuristic algorithms in optimization. *Journal of modeling in engineering*, *12*(38), 27-43. (**In Persian**). https://modelling.semnan.ac.ir/article_1677.html?lang=en

Shayeghi, H., & Dadashpour, J. (2012). Anarchic society optimization based PID control of an automatic voltage regulator (AVR) system. *Electrical and electronic engineering*, *2*(4), 199-207.

Shi, Y. (2015). An optimization algorithm based on brainstorming process. In *emerging research on swarm intelligence and algorithm optimization* (pp. 1-35). IGI Global.

Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, *12*(6), 702-713.

Tabari, A., & Ahmad, A. (2017). A new optimization method: Electro-Search algorithm. *Computers & chemical engineering*, *103*, 1-11.

Wang, T., & Yang, L. (2018). Beetle swarm optimization algorithm: Theory and application. arXiv preprint arXiv:1808.00206

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

Yaghini, M., & Akhavan Kazemzadeh, M. R. (2016). *Meta-innovation optimization algorithms*. Jihad Daneshgahi Publications, Amir Kabir Industrial Branch

Yan, G. W., & Hao, Z. J. (2013). A novel optimization algorithm based on atmosphere clouds model. *International journal of computational intelligence and applications*, *12*(01), 1350002. https://doi.org/10.1142/S1469026813500028

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Berlin, Heidelberg: Springer.

Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, *3*(1), 24-36.

Zhang, L. M., Dahlmann, C., & Zhang, Y. (2009, November). Human-inspired algorithms for continuous function optimization. In *2009 IEEE international conference on intelligent computing and intelligent systems* (pp. 318-321). IEEE.