بهینه‌سازی مسائل مهندسی با قیود گسسته و پیوسته به‌کمک الگوریتم‌های فراابتکاری تطبیقی پویا

سیدمصطفی موسوی[1]، علیرضا طلوعی[*2] ، محمدحسین طلوعی[3]

۱–دانشجوی دکتری مهندسی هوافضا دانشگاه شهید بهشتی تهران، ۲– دانشیار دانشگاه شهید بهشتی تهران۳– دانشجوی کارشناسی ارشد

چکیده

هدف این مقاله، پیاده‌سازی و بررسی روش‌های مختلف الگوریتم‌های فراابتکاری برای حل پنج مسأله مهندسی است. لذا این مسائل مهندسی، به صورت مقید، بررسی شده سپس با استفاده از پنج الگوریتم فراابتکاری کرم شب‌تاب، رقابت استعماری، قورباغه، مورچگان و گرگ خاکستری و با هدف کاهش هزینه‌های مسائل مهندسی، بهینه‌سازی شده و نتایج حل آن صحت‌سنجی شده است. در واقع، در هر یک از الگوریتم‌های بهینه‌سازی استفاده شده، یک عامل تطبیقی پویا برای متعادل‌کردن نرخ همگرایی و توانایی جستجوی بهینه مطلق با تنظیم سرعت جستجو در طول فرآیند جستجو معرفی شده است. بررسی‌ها نشان می‌دهد که در هر یک از الگوریتم‌ها، تکنیک‌هایی برای خروج از بهینه محلی استفاده می‌شود که باعث می‌شود پاسخ‌ها به بهینه مطلق همگرا شوند. برای ارزیابی کیفیت و دقت الگوریتم‌ها، از تست حساسیت سنجی و مقایسه اعداد همگرایی برای نتایج حاصل از اجرای هر الگوریتم بر روی داده‌ها استفاده شده است. نتایج به‌دست آمده نشان می‌دهد که الگوریتم کرم شب‌تاب در مساله فنرکششی، الگوریتم قورباغه در مساله خرپای سه میله‌ای، الگوریتم رقابت استعماری در مسائل کاهنده سرعت و طراحی چرخدنده، و الگوریتم گرگ خاکستری در مساله مخزن تحت فشار، در یافتن بهینه مطلق، عملکرد دقیق‌تری را ارائه کردند. در واقع این الگوریتم‌ها برای حل بهینه مسائل با تولید جمعیت تصادفی، ایجاد همسایگی و انتخاب بهترین همسایه به شرط ارضاء قیود متغیرهای مسائل، رسیدن به حل بهینه را آسان می‌کند. در نتیجه، این مقاله نشان می‌دهد که هر الگوریتم فراابتکاری می‌تواند در یک مسئله مهندسی مشخص، بسته به نوع مساله و شرایط محیطی، عملکرد بهتری داشته باشد.

کلیدواژه‌ها: مسائل بهینه‌سازی طراحی مهندسی مقید، متغیرهای طراحی پیوسته و گسسته، فراابتکاری، الگوریتم بهینه‌سازی رقابت استعماری، مورچگان، قورباغه، گرگ خاکستری، کرم شب‌تاب

Optimization of engineering problems with discrete and continuous constraints using dynamic adaptive meta-heuristic algorithms

SayedMostafa Mousavi, Alireza Toluei *

ShahidBeheshti University

(Received: , Accepted:)

Abstract

The purpose of this article is to implement and review different methods of meta-heuristic algorithms to solve five engineering problems. Therefore, these engineering problems were examined in a restricted manner, then using five meta-heuristic algorithms of firefly, colonial competition, frog, ants and gray wolf, and with the aim of reducing the costs of engineering problems, they were optimized and the results of its solution were verified. In fact, in each of the used optimization algorithms, a dynamic adaptive factor is introduced to balance the convergence rate and absolute optimal search ability by adjusting the search speed during the search process. Investigations show that in each of the algorithms, techniques are used to leave the local optimum, which makes the answers converge to the absolute optimum. To evaluate the quality and accuracy of the algorithms, the sensitivity test and the comparison of the convergence numbers for the results of the implementation of each algorithm on the data have been used. The obtained results show that the firefly algorithm in the spring tension problem, the frog algorithm in the three-bar truss problem, the colonial competition algorithm in the speed reducer and gear design problems, and the gray wolf algorithm in the pressure tank problem have a more accurate performance in finding the absolute optimum. In fact, these algorithms make it easy to achieve the optimal solution by generating a random population, creating a neighborhood and choosing the best neighbor, provided that the constraints of the variables of the problem are satisfied. As a result, this paper shows that any meta-heuristic algorithm can perform better in a specific engineering problem, depending on the type of problem and environmental conditions.

Keywords: Constrained engineering design optimization problems, continuous and discrete design variables, meta-heuristics, Imperialist Competitive, ants, Frog, grey wolf , firefly.

*Corresponding Author :Toloei@ihu.ac.ir

۱. مقدمه

تا به‌حال الگوریتم‌های بهینه‌سازی بی‌شماری جهت حل مسائل متنوع بهینه‌سازی طراحی مهندسی پیشنهاد شده‌اند، مسائلی‌که معمولا شامل قیود غیرخطی می‌شوند.

الگوریتم‌های بهینه‌سازی را می‌توان تقریبا به دو دسته تقسیم کرد: الگوریتم تصادفی و الگوریتم معین. روش‌های سنتی بهینه‌سازی معین، مانند روش شیب نزولی[1]، روش شبه نیوتنی و روش نیوتن بازگشتی داخلی، که معمولا مبتنی بر گرادیان هستند و الگوریتم‌ها و شرایط متفاوت توابع هدف مورد نیاز است.

این روش‌ها برای مسائل بهینه‌سازی پیچیده و غیرخطی با ابعاد بالا، به‌ویژه زمانی‌که توابع و قیود هدف ناپیوسته هستند، ناکارآمد و نادقیق هستند[۱].

الگوریتم‌های بهینه‌سازی تصادفی متعددی مانند الگوریتم بهینه‌سازی ازدحام ذرات[۲][2]، الگوریتم ژنتیک[۴و ۳] ، الگوریتم کرم شب‌تاب[۶]، بهینه‌سازی کلونی مورچگان[3] [۷]، کلونی زنبورهای مصنوعی[۲۴،۸و۲۵]،قورباغه[۹][4]، رقابت استعماری[۱۰][5]، الگوریتم بهینه‌سازی گرگ خاکستری[۵] [6] برای غلبه بر این موارد پیشنهاد شده است. این الگوریتم‌های بهینه‌سازی تصادفی معمولا فراابتکاری هستند و از پدیده‌های فیزیکی و طبیعی الهام می‌گیرند.

مطابق مقالات دوریگو [7]الگوریتم بهینه‌سازی مورچگان الهام گرفته از رفتار مورچگان در جستجوی منابع غذایی است و در سال ۱۹۹۱ توسعه یافته است.

این الگوریتم بهبودیابی، مبتنی بر جمعیت است که به صورت متوالی بر روی یک مسئله بهینه‌سازی عمل می‌کند و قدرت بالایی در بهینه‌سازی مسائل پیچیده دارد. لذا با استفاده از اطلاعات جمع‌آوری شده توسط مورچگان، قادراست به راه‌حل‌های بهینه نزدیک شود و با تکرار این فرآیند، بهبودهای اضافی را نیز ایجاد کند[۷].

همچنین میرجلیلی در سال۲۰۱۴، الگوریتم بهینه‌سازی گرگ خاکستری را مطرح نمود، که الهام گرفته از رفتار اجتماعی گرگ‌های خاکستری در طبیعت است. این الگوریتم برای حل مسائل بهینه‌سازی و جستجوی فضاهای چندمعیاره استفاده می‌شود و قابلیت همگرایی سریع و جستجوی مؤثر را داراست و می‌تواند بهترین حل ممکن را در فضای جستجوی بزرگ و پیچیده پیدا کند.

با اجرای مراحل تعامل و بهبود مکان گرگ‌ها، الگوریتم قادر است به نتایج بهتر و بهینه‌تر دست یابد و در مسائل پیچیده‌تر نیز عملکرد خوبی داشته باشد[۱۱و۵].

در سال ۲۰۰۷ نیز گارگاری و لوکاس[8] الگوریتم رقابت استعماری را ارائه نمودند که یک الگوریتم فراابتکاری، الهام گرفته از رفتار رقابت و همکاری بین جامعه‌های استعماری در طبیعت است. این الگوریتم برای حل مسائل بهینه‌سازی و جستجوی فضاهای چندمعیاره مورد استفاده قرار می‌گیرد[۱۳،۱۲] . در سال ۲۰۰۳ نیز، مظفر و همکاران، الگوریتم قورباغه را ارائه نمودند که بر پایه رفتار قورباغه‌ها در محیط طبیعی برای حل مسائل بهینه‌سازی مورد استفاده قرار گرفته شده است. پس از آن نیز در بسیاری از مسائل مهندسی مکانیک با هدف کاهش هزینه‌ها استفاده شده است [۱۴، ۱۵و۱۶].

از طرفی درسال ۲۰۰۵، الگوریتم کلونی زنبورهای مصنوعی توسط کارابوگا[9]، برای بهینه‌سازی پارامترهای مسأله ارائه شد. این الگوریتم به‌طرز مشابه با رفتار و سازماندهی کلونی‌های زنبورهای عسل در طبیعت برای حل مسائل بهینه‌سازی استفاده شده است [۲۵و۲۶].

درمیان همه این الگوریتم‌های بهینه‌سازی تصادفی، الگوریتم بهینه‌سازی کرم شب‌تاب[10]به‌طورگسترده برای حل مسائل بهینه‌سازی مختلف استفاده می‌شود و برای اولین بار در سال ۲۰۰۷ توسط این-شی یانگ[11]معرفی و در محاسبات مهندسی کارآمد نشان داده شده است. از ویژگی‌های آن پیاده‌سازی آسان، و قابل اعتماد در جستجوی بهینه مطلق است[۱۷،۶و۱۸].

دراین مقاله از الگوریتم کرم شب‌تاب استفاده شده که یک الگوریتم بهینه‌سازی نشأت گرفته شده از رفتار گروهی کرم‌ها است و برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد[۲۰،۱۹] .

این الگوریتم براساس رفتار اجتماعی و تعامل بین‌کرم‌ها در محیط طبیعی الهام گرفته شده است. از آنجا که مسائل هوافضایی به‌دلیل پیچیدگی و ابعاد بالا، نیازمند الگوریتم‌های بهینه‌سازی قوی هستند، الگوریتم کرم شب‌تاب به عنوان یکی از ابزارهای قدرتمند در این حوزه استفاده می‌شود. این الگوریتم به‌طور گسترده درمسائل مربوط به علوم هوافضا مورد استفاده قرارگرفته است[۲۱و۲۲].

۲. روش‌های حل

الگوریتم‌های فراابتکاری یک مجموعه از روش‌ها و الگوریتم‌های پیشرفته در حوزه بهینه‌سازی و جستجوی مسائل پیچیده هستند. این الگوریتم‌ها بر اساس الهام از فرآیندها و رفتارهای طبیعی و اجتماعی مانندجستجوی غذا در جوامع حیوانی، پدیدار شده‌اند. هدف اصلی الگوریتم‌های فراابتکاری، یافتن بهینه‌ترین راه حل‌ها در فضای جستجو و خروج از نقاط بهینه محلی می‌باشد.

بعضی از این روش‌ها، مثلا روش کلونی مورچگان مناسب برای مسائل با متغیرهای گسسته هستند. پنج روش حلی که برای این مقاله

---

[1] Steepest Descend
[2] PSO : Particle Swarm Optimization
[3] Ant Colony Optimization Algorithm
[4] Shuffled Frog Leaping Algorithm
[5] Imperialist Competitive Algorithm
[6] Grey Wolf Optimizer
[7] Marco Dorigo
[8] E Atashpaz - Gargari & C Lucas
[9] Karaboga
[10] Firefly Algorithm
[11] Xin-She Yang

انتخاب شده‌اند، برای مسائل با متغیرهای پیوسته، گسسته و یا ترکیب آنها مناسب هستند.

۱-۲. الگوریتم گرگ خاکستری

این الگوریتم در سال ۲۰۱۴ توسط سید علی میرجلیلی ارائه شده است. الگوریتم گرگ خاکستری یک الگوریتم متاهیورستیک است که از ساختار سلسله مراتبی[1] و رفتار اجتماعی گرگ‌های خاکستری در هنگام شکار کردن الهام گرفته است. در الگوریتم گرگ خاکستری، جمعیتی از گرگ‌ها به‌صورت تصادفی در فضای جستجو قرار می‌گیرند. هر گرگ مشخصه‌هایی مانند موقعیت، سرعت و شکار موفقیت‌آمیز خود را دارد. این مشخصه‌ها بر اساس قوانینی که بر اساس رفتار گروهی گرگ‌ها تعیین شده‌اند، تغییر می‌کنند. در هر مرحله از الگوریتم، گرگ‌ها به دو عملکرد اصلی تکیه می‌کنند: اکتشاف و بهره‌برداری. اکتشاف به معنای جستجو در مناطق جدید فضای جستجو و کشف نقاط بهتر است، در حالی که بهره‌برداری به معنای بهره‌برداری از نقاط قوی و بهترین‌هاست[۵].

در الگوریتم گرگ خاکستری، گروه گرگ‌ها با توجه به رفتارشان در شکار و اجتماع، به چهار نوع تقسیم می‌شوند. این انواع شامل موارد زیر هستند:

گرگ آلفا[2] : گرگ آلفا به عنوان رهبر گروه عمل می‌کند و معمولا بهترین عملکرد را دارد. موقعیت آلفا به عنوان بهترین حل موجود در جمعیت در نظر گرفته می‌شود و بقیه گرگ‌ها بر اساس آن بهبود می‌یابند.

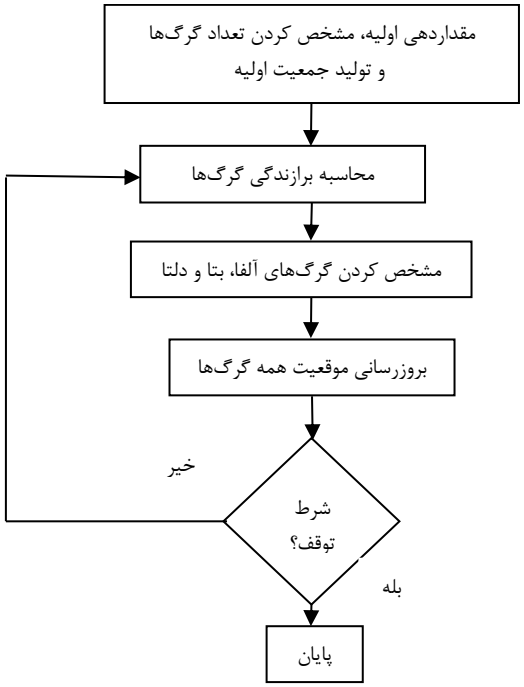
گرگ بتا[3]: گرگ بتا مسئول کمک به گرگ آلفا در جستجوی بهترین حل است. معمولا در نزدیکی گرگ آلفا قرار می‌گیرد و سعی می‌کند بهترین راهنمایی‌ها را در جهت بهبود عملکرد ارائه دهد.

گرگ دلتا[4] : گرگ دلتا نقشی مشابه گرگ بتا دارد و به گرگ آلفا و بتا در جستجوی بهترین حل کمک می‌کند. در نزدیکی گرگ بتا قرار می‌گیرد و تلاش می‌کند با تجمیع راهنمایی‌های مختلف بهبودی در عملکرد داشته باشد.

گرگ اپسیلون[5]: گرگ اپسیلون به عنوان گرگی عادی در جمعیت حضور دارد و بر اساس رفتار سایر گرگ‌ها تغییر می‌کند. این گرگ‌ها به عنوان جستجوگران اصلی در جستجوی مناطق جدید و اکتشافی عمل می‌کنند[۱۱].

۱-۱-۲. روندنمای الگوریتم گرگ های خاکستری

در شکل۱ روندنمای کلی الگوریتم گرگ های خاکستری مشاهده می‌شود.



شکل ۱. روندنمای الگوریتم بهینه‌سازی گرگ خاکستری [۵]

۲-۲. الگوریتم کلونی مورچگان

این الگوریتم یک الگوریتم فراابتکاری بر مبنای رفتار و ارتباطات مورچگان در جستجوی غذا می‌باشد و در دهه ۱۹۹۰ توسط مارک دوریگو و همکاران توسعه داده شد و بر اساس تعاون مورچگان در سفرهای غذاگیری الهام گرفته شده است.[۷]

در الگوریتم کلونی مورچگان، مورچگان به‌صورت تصادفی در فضای جستجو قرار می‌گیرند و مواد شیمیایی به نام فرومون‌ها را رها می‌کنند. این فرومون‌ها ارتباطاتی بین مورچگان ایجاد می‌کنند و به عنوان راهنمایی برای انتخاب مسیرها و جستجوی بهینه عمل می‌کنند. درحقیقت الگوریتم کلونی مورچگان از رفتار واقعی مورچه‌ها الهام گرفته شده است، اما شبیه‌سازی کاملی از آن وجود ندارد، اما به عنوان یک ابزار قدرتمند بهینه‌سازی برای حل طیف گسترده‌ای از مسائل در دنیای واقعی به اثبات رسیده است. برخی از کاربردهای برجسته این الگوریتم در بهینه‌سازی مسیر، طراحی مدار، برنامه‌ریزی وسایل نقلیه و غیره است.

در هر مرحله از الگوریتم، مورچگان به دو عملکرد اصلی تکیه می‌کنند: اکتشاف و بهره‌برداری[۷].

به طور کلی، الگوریتم کلونی مورچگان شامل مراحل زیر است:

۱- مقداردهی اولیه: مقداردهی اولیه به مورچگان و فرومون‌ها صورت می‌گیرد.

---

[1] Hieratical

[2] Alpha

[3] Beta

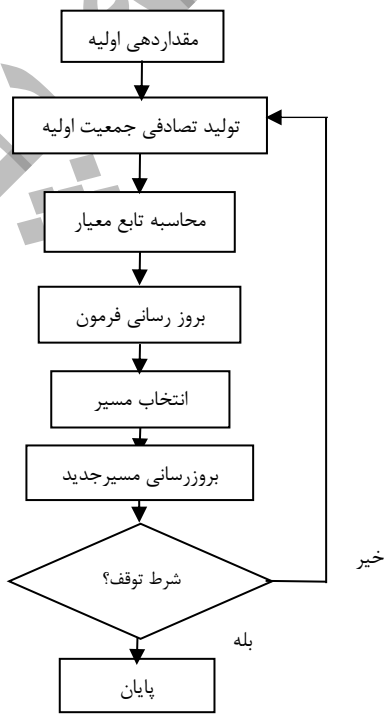
[4] Delta

[5] Epsilon

۲- حرکت مورچگان: مورچگان به صورت تصادفی در فضای جستجو حرکت می‌کنند.
۳- ترکیب فرومون‌ها: مقدار فرومون‌ها بر روی مسیرهایی که مورچگان طی کرده‌اند، به روزرسانی می‌شود.
۴- انتخاب مسیر: مورچگان بر اساس مقدار فرومون‌ها و احتمال‌های مرتبط با مسیرها، مسیرهای بهینه را انتخاب می‌کنند.
۵- به‌روزرسانی بهترین مسیر: بهترین مسیرهایی که تاکنون توسط مورچگان پیدا شده‌اند، به عنوان بهترین حل موجود در نظر گرفته می‌شودو به‌روزرسانی می‌شوند.
۶- تا زمان اتمام شرایط خاتمه: مراحل ۲ تا ۵ تکرار می‌شوند تا به شرایط خاتمه برسند[۷].

۲-۲-۱. روندنمای الگوریتم کلونی مورچگان

در شکل۲ نیز روندنمای کلی الگوریتم کلونی مورچگان مشاهده می‌شود.



شکل ۲.روندنمای الگوریتم بهینه‌سازی کلونی مورچگان [۳۰]

۲-۳. معرفی الگوریتم بهینه‌سازی کرم شب‌تاب

الگوریتم فراابتکاری کرم شب‌تاب براساس مکانیزم‌های تعاملی کرم‌های شب‌تاب در جستجوی منابع نور الهام گرفته شده است.

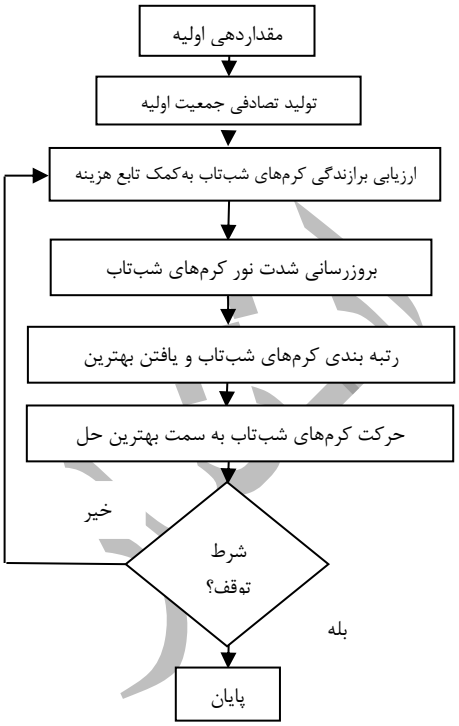
در این الگوریتم، کرم‌های شب‌تاب به‌صورت تصادفی در فضای جستجو قرار می‌گیرند و منابع نور را توزیع می‌کنند. هر کرم شب‌تاب قادر به تشخیص میزان نور در اطراف خود است و با توجه به میزان نور واکنش‌هایی انجام می‌دهد[۲۲].

به طور کلی، الگوریتم کرم شب‌تاب شامل مراحل زیر است:

۱- مقداردهی اولیه: مقداردهی اولیه به کرم‌های شب‌تاب و منابع نور صورت می‌گیرد.
۲- حرکت کرم‌ها: کرم‌های شب‌تاب به‌صورت تصادفی در فضای جستجو حرکت کرده و نور در اطراف خود را اندازه‌گیری می‌کنند.
۳- ارتباطات کرم‌ها: کرم‌ها با توجه به میزان نور در اطراف خود و بر اساس قوانین تعاملی، واکنش‌هایی انجام می‌دهند. این واکنش‌ها شامل جذب منابع نور و جلوگیری از رسیدن کرم‌های دیگر به منابع نور است.
۴- به‌روزرسانی شعاع نور: بر اساس واکنش‌ها و میزان نور در اطراف، شعاع نور هر کرم شب‌تاب به‌روزرسانی می‌شود.
۵- به‌روزرسانی منابع نور: بر اساس شعاع نور کرم‌ها و توزیع آنها در فضای جستجو، منابع نور به‌روزرسانی می‌شوند.
۶- تا زمان اتمام شرایط خاتمه: مراحل ۲ تا ۵ تکرار می‌شوند تا به شرایط خاتمه مانند تعداد تکرارها یا دستیابی به یک حل بهینه برسد.[۲۲].

۲-۳-۱. روندنمای الگوریتم کرم شب‌تاب

در شکل۳ روندنمای الگوریتم کرم شب تاب را مشاهده می‌کنیم.



شکل ۳. فلوچارت الگوریتم بهینه‌سازی کرم شب‌تاب[۲۳]

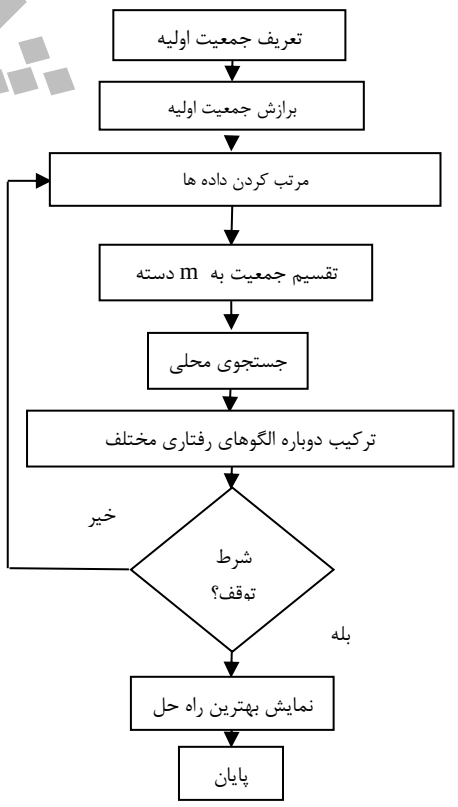
۲-۴. معرفی الگوریتم بهینه سازی قورباغه

الگوریتم قورباغه یک الگوریتم بهینه سازی جهش قورباغه[1]، یکی ازالگوریتم های بهینه سازی فرا ابتکاری است که از رفتار اجتماعی قورباغه ها در طبیعت الهام گرفته شده است،[۱۴،۱۵و۱۶]

الگوریتم بهینه‌سازی قورباغه الگوریتم فراابتکاری است که بر پایه رفتار قورباغه‌ها در جستجوی غذا توسعه یافته است. این الگوریتم برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. در این الگوریتم، یک جمعیت از قورباغه‌ها (که هر قورباغه یک حالت ممکن را نشان می‌دهد) ایجاد می‌شود. این قورباغه‌ها در جستجوی یافتن یک حل بهینه برای مسئله قرار می‌گیرند.الگوریتم قورباغه به دلیل رفتار مشابه قورباغه در حرکت و جستجو در فضای سه‌بعدی، برای حل مسائل پیچیده و چندمعیاره کارآمد است. این الگوریتم می‌تواند در مسائل بهینه‌سازی، هوش مصنوعی، بهینه‌سازی ترکیبی و سایر حوزه‌های مرتبط مورد استفاده قرار گیرد[۹].

۲-۴-۱. روندنمای الگوریتم قورباغه

در شکل۴ روندنمای الگوریتم قورباغه را مشاهده می‌کنیم.



شکل ۴. فلوچارت الگوریتم بهینه‌سازی قورباغه[۱۴]

روند کار الگوریتم قورباغه به شرح زیر است:

۱- اولیه: جمعیتی از قورباغه‌ها به صورت تصادفی ایجاد می‌شود.

۲- ترکیب: قورباغه‌ها بر اساس مقدار ارزیابی، به صورت تصادفی در گروه‌های کوچکتری تقسیم می‌شوند. گروه‌ها با توجه به مقدار ارزیابی از بهترین تا بدترین مرتب می‌شوند.

۳- بهبود: در هر گروه، قورباغه‌ها با استفاده از روش بهبود محلی وضعیت خود را بهبود می‌بخشند.

۴- حرکت: قورباغه‌ها در هر گروه به صورت تصادفی حرکت می‌کنند و وضعیت جدیدی را برای خود ایجاد می‌کنند.

۵- ادغام: پس از حرکت، گروه‌های قورباغه‌ها ادغام می‌شوند و جمعیت جدیدی تشکیل می‌دهند.

۶- به‌رورسانی: با توجه به مقدار ارزیابی، قورباغه‌ها به‌روزرسانی می‌شوند و بهترین حالت‌ها دریافت می‌شوند.

۷- تا زمان اتمام شرایط خاتمه: مراحل ۳ تا ۷ تکرار می‌شوند تا به شرایط خاتمه برسیم.

۸- ارزیابی: هر قورباغه وضعیت خود را ارزیابی کرده و مقدار مناسبی برای آن اختصاص می‌دهد[۱۵و۱۶].

۲-۵. معرفی الگوریتم بهینه سازی رقابت استعماری

الگوریتم بهینه‌سازی رقابت استعماری یک الگوریتم محاسباتی است که بر اساس مفهوم رقابت و هم‌اندیشی در یک سازمان استعماری تشکیل شده است.[۱۳،۲۷] این الگوریتم برای حل مسائل بهینه‌سازی پیچیده و چندمعیاره استفاده می‌شود. الگوریتم بهینه‌سازی رقابت استعماری از ترکیب الگوریتم‌های تکاملی و الگوریتم‌های مبتنی بر رقابت بین گروه‌ها بهره می‌برد[۱۲].

روند کار الگوریتم بهینه‌سازی رقابت استعماری به شرح زیر است:

۱- مقداردهی اولیه: استعمارها و جانداران به صورت تصادفی در فضای جستجو قرار می‌گیرند.

۲- ارزیابی: هر جانداران وضعیت خود را ارزیابی کرده و مقدار مناسبی برای آن اختصاص می‌دهد.

۳- تعامل استعمارها: استعمارها به صورت مکرر اطلاعات و تجربیات خود را با استفاده از عملکرد جانداران مبادله می‌کنند.

۴- رقابت و انتخاب: جانداران درون هر استعمار به رقابت با یکدیگر در جهت بهبود مقدار ارزیابی خود می‌پردازند.. جانداران با برتری در رقابت در استعمار خود، به نسل بعدی منتقل می‌شوند.

۵- خاتمه: تا زمان اتمام کار، مراحل ۳و۴ تکرار می‌شوند.

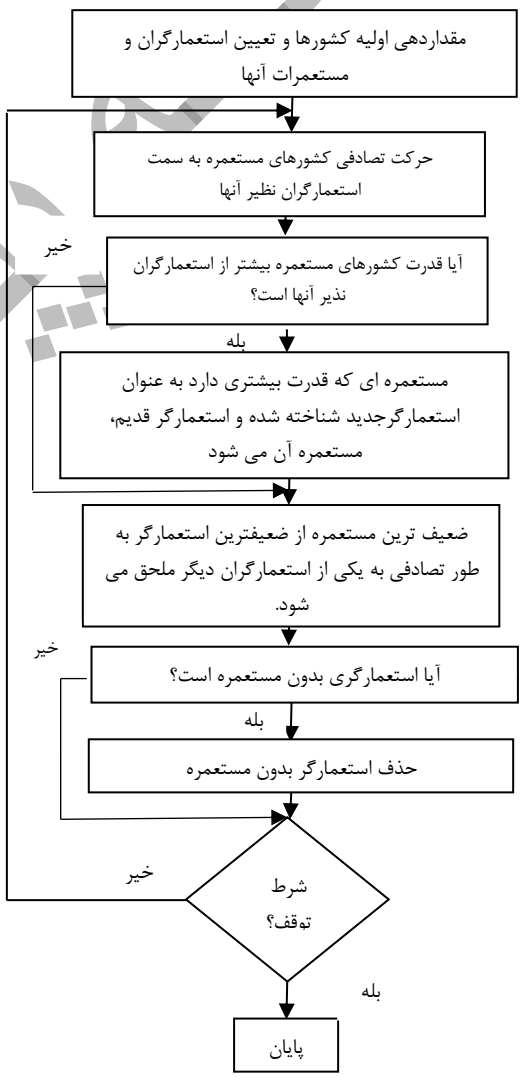
۶- خروجی: پس از پایان مراحل الگوریتم، بهترین حل یافت شده به عنوان خروجی الگوریتم انتخاب می‌شود. پس از پایان الگوریتم، مقدار بهینه‌سازی شده تابع هدف و مقدار

بهینه‌یابی شده برای متغیرها به عنوان خروجی الگوریتم ارائه می‌شود.

در واقع الگوریتم بهینه‌سازی رقابت استعماری به دلیل قابلیت تطبیق با مسائل پیچیده و چندمعیاره و قدرت استفاده از رقابت و هم‌اندیشی بین امپراتورها، به عنوان یک روش قوی و مؤثر در حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرد. الگوریتم بهینه‌سازی رقابت استعماری قابلیت تعمیم و انعطاف‌پذیری دارد و می‌تواند در مسائل مختلفی از جمله بهینه‌سازی تابع، بهینه‌سازی مسئله‌های ترکیبی و بهینه‌سازی پارامترها مورد استفاده قرارگیرد[۱۰].

۱-۵-۲. روندنمای الگوریتم رقابت استعماری

شکل۵ روندنمای الگوریتم قورباغه را نشان می‌دهد.



شکل ۵. فلوچارت الگوریتم بهینه‌سازی رقابت استعماری[۲۷]

۶-۲. عامل تطبیقی پویا

در واقع عامل تطبیقی پویا به‌منظور متعادل‌کردن نرخ همگرایی و توانایی جستجوی بهینه مطلق در طول فرآیند جستجو معرفی شده است. این عامل توانایی تطبیق سرعت جستجو را در محیط مسئله دارد و با تغییر سرعت جستجو و تطبیق آن با شرایط محیطی، به‌دنبال بهینه‌سازی بهتری می‌گردد. با این تطبیق پویا، الگوریتم‌ها قادر به بهبود نرخ همگرایی و جستجوی بهینه‌تر در فضای جستجوی خود هستند. در این مقاله، با استفاده از تکنیک‌های خروج از بهینه محلی، این الگوریتم‌ها توانایی جستجو در فضای جستجوی گسترده‌تر را دارند و می‌توانند به‌نتایج بهینه مطلق نزدیک شوند.

در الگوریتم گرگ خاکستری، عامل تطبیقی پویا نقش مهمی در بهبود عملکرد الگوریتم ایفا می‌کند. این عامل معمولا برای تطبیق نرخ اکتشاف و بهبود جستجو استفاده می‌شود و بستگی به پارامترهای مرتبط با نرخ اکتشاف و تغییرات محیطی دارد، رابطه(۱) بخوبی نمایانگر این عامل است:

$$a = 2 - t * \left(\frac{(2)}{max_iter}\right) \qquad (۱)$$

در این رابطه، a نرخ اکتشاف است. t زمان فعلی یا تکرار فعلی است و max_iterحداکثر تعداد تکرارها است.

با استفاده رابطه(۱)، عامل تطبیقی پویا می‌تواند نرخ اکتشاف را در طول فرآیند جستجو تنظیم کند. درابتدای جستجو این نرخ افزایش و جستجو تصادفی و منجر به خروج از اکسترمم‌های محلی شده و با گذشت زمان و تکرارها، نرخ اکتشاف کاهش می‌یابد و الگوریتم از جستجوی بیشتری در نزدیکی به مکان‌های بهینه استفاده می‌کند. لذا این تطبیق باعث می‌شود الگوریتم بهبود یابد و نتایج بهینه‌تری را تولید کند.

در الگوریتم کلونی مورچگان نیز عامل تطبیقی پویا معمولا برای تطبیق نرخ انتشار فرومون استفاده می‌شود. دراین الگوریتم، یکی از روابط کلیدی که توسط عامل تطبیقی پویا تغییر می‌کند، رابطه انتشار فرومون است. این‌رابطه به شکل زیر بیان شده است:

$$P_clone(i) = P_min + (P_max - P_min) * \frac{(fit(i) - fit_min)}{(fit_max - fit_min)} \qquad (۲)$$

در رابطه(۲)، P_clone(i) انتشار فرومون در جهت i است.

P_min و P_max حداقل وحداکثر مقادیر انتشار فرومون هستند.

fit(i) مقدار تابع ارزیابی (fitness function) برای انتشار فرومون است و fit_min و fit_max حداقل و حداکثر مقادیر تابع ارزیابی در میزان انتشار فرومون هستند.

در واقع عامل تطبیقی پویا با تطبیق مقدار P_clone(i) با شرایط محیطی و مقادیر تابع ارزیابی، انتشار فرومون را تنظیم می‌کند.

لذا مسیرهایی‌که توسط مورچه‌های بیشتردنبال می‌شوند، با غلظت فرومون بیشتری علامت‌گذاری می‌شوند و از این طریق امکان یافتن

بهینه‌ترین حل و هدایت مورچه‌ها به‌سمت منابع غذایی را فراهم می‌کند.

در الگوریتم کرم شب‌تاب هم، فاکتور تصادفی سازی α به‌عنوان عامل پویا در طی فرایند جستجو تعدیل می‌شود. در اینجا k بین ۰ و ۱ است و نرخ افزایشی نام دارد که می‌تواند سرعت کاهش α را تعدیل کند، رابطه(۳) نمایانگر این عامل تطبیقی پویا است:

$$\alpha(t+1) = \alpha(t).\exp(-k.\frac{t}{G_{max}}) \quad (۳)$$

نرخ کاهشی k ممکن است روی کارایی پارامتر مدل تاثیر بگذارد. درارابطه(۳)، G_{max} حداکثر تعداد تعداد تکرار و t نشان دهنده شاخص تولید نور می‌باشد.

درادامه در الگوریتم قورباغه نیز عامل تطبیقی پویا برای بهبود نرخ همگرایی و جستجو در فضای جستجوی الگوریتم استفاده می‌شود.

دراین الگوریتم، از رابطه(۴) برای تغییر سرعت جستجو استفاده می‌شود:

$$V(t+1) = V(t) + A(t) * (X_best - X(t)) \quad (۴)$$

در این رابطه، V(t+1) سرعت جستجو در زمان t+1 است.

V(t) سرعت جستجو در زمان t است.

A(t) عامل تطبیقی پویا است که مقدار آن در طول فرآیند جستجو تغییر می‌کند. X_best بهترین حل یافت شده تا آن لحظه است.

وX(t) مکان فعلی جستجو در زمان t است.

در حقیقت عامل تطبیقی پویا با تطبیق مقدار A(t) با شرایط محیطی، سرعت جستجو را تنظیم می‌کند. مقدار A(t) می‌تواند به صورت ثابت، تصادفی یا با استفاده از روش‌های خاص تعیین شود. تغییر مقدار A(t) در طول فرآیند جستجو، به این الگوریتم قابلیت تطبیق با محیط و بهبود نرخ همگرایی را می‌دهد[۲۸].

درآخردر الگوریتم رقابت استعماری عامل تطبیقی پویا نقش مهمی در تطبیق و بهبود عملکرد الگوریتم ایفا می‌کند. این عامل معمولا برای تطبیق نرخ مهاجرت و تغییر مقدار آن در طول فرآیند جستجو استفاده می‌شود. رابطه‌ای‌که توسط عامل تطبیقی پویا در الگوریتم رقابت استعماری تأثیر دارد، بستگی به پارامترهای مرتبط با مهاجرت و تغییرات محیطی دارد. رابطه(۵) از عامل تطبیقی پویا در این الگوریتم استفاده می‌کند:

$$Migration_rate(t) = Migration_rate_initial*e^{(-alpha*t)} \quad (۵)$$

در این رابطه،Migration_rate(t) نرخ مهاجرت در زمان t است.

عبارت Migration_rate_initial نرخ مهاجرت اولیه است.

e نماد عدد اویلر است.

alpha پارامتری‌است که توسط عامل تطبیقی پویا تنظیم می‌شود و t زمان فعلی در الگوریتم است. با استفاده رابطه(۵)، عامل تطبیقی پویا می‌تواند نرخ مهاجرت را در طول فرآیند جستجو تنظیم کند. با گذشت زمان، نرخ مهاجرت کاهش می‌یابد و تأثیر مهاجرت بین جمعیت‌ها کاهش می‌یابد. این تطبیق باعث می‌شود که الگوریتم بهبود یابد و به جمعیت‌هایی که در نتیجه‌ی عملکرد بهتری هستند، اجازه بدهد که بیشتر منتشر شوند و تأثیر بیشتری در روند جستجو داشته باشند.

لذا عامل تطبیقی پویا در الگوریتم‌های تکاملی کلونی مورچگان، قورباغه، گرگ خاکستری، رقابت استعماری و کرم شب‌تاب با افزایش خود درابتدای جستجو می‌تواند موجب خروج از بهینه محلی، عملکرد بهتر و جستجوی تصادفی و گسترده‌تر در فضای جستجوی الگوریتم‌ها شده و با کاهش تدریجی به‌رسیدن به نقاط بهینه‌کمک کنند.

۳. معرفی مسائل بهینه سازی مهندسی

۳-۱. معرفی مسائل بهینه سازی مهندسی مقید

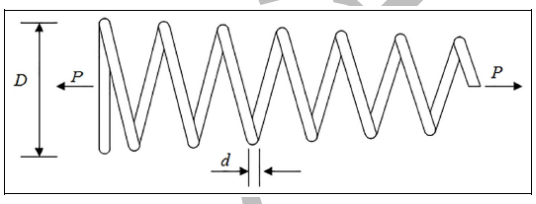
در این بخش، پنج مسأله از مسائل بهینه‌سازی مهندسی مقید شده که دارای توابع هدف، متغیرهای طراحی و قیود مختلف هستند، انتخاب می‌شوند و عملکرد الگوریتم کرم شب‌تاب، رقابت استعماری و بقیه الگوریتم‌های پیشنهادی از نظر کیفیت راه‌حل و پایداری و همچنین نرخ هم‌گرایی مورد بررسی قرار می‌گیرد. این پنج مسئله بهینه سازی مهندسی مقید با توجه به‌دسته بندی متغیرهایشان به مسائل بهینه‌سازی پیوسته و گسسته تقسیم می‌شوند.

نتایج و بهترین راه‌حل‌های همه الگوریتم‌ها به‌کمک حساسیت-سنجی پارامترها، برای این مسائل بهینه‌سازی مهندسی، در بیش از ۳۰ اجرای مستقل به‌دست می‌آیند. جداول این بخش شامل پنج الگوریتم بخش۲ این مقاله و الگوریتم‌های موجود در مرجع[۲۹]است که با رنگ خاکستری در ردیف‌های انتهایی جداول مشخص شده است.

۳-۱-۱. مسائل بهینه سازی مقید با متغیرهای پیوسته

۳-۱-۱-۱. مساله طراحی فنر کششی/فشردهی

شکل شماره ۶ شماتیک یک فنر کششی / فشرده را نشان می‌دهد. هدف طراحی مسئله کشش/ فشردگی به‌حداقل رساندن وزن آن با



۶. شماتیک فنرکششی[۳۰]

اعمال قیود حداقل تغییر شکل، تنش برشی، فرکانس موج و حداکثر قطر خارجی است، بطوریکه تغییر شکل در فنرها معمولا به قانون هوک(نسبت نیروی بازگرداننده وارده از سوی فنر به میزان تغییر شکل فنر برابر است با مقدار ثابت فنر یا k) مرتبط است. فرکانس موج نیز معمولا با توجه به ویژگی‌های مکانیکی فنر، از جمله سختی (k) و جرم (m) سیستم محاسبه می‌شود.

در واقع این قیود، چهار معادله نابرابر غیرخطی را تشکیل می‌دهند که در رابطه۶ شرح داده شده است. مسئله طراحی دارای سه متغیر طراحی شامل قطر سیم d، قطر متوسط سیم پیچD، و تعداد سیم

پیچ‌های فعال N، است که در تابع هدف$J(x)$ و تابع قیود به ترتیب با x1، x2 و x3 نشان داده شده است. در واقع تعداد دور سیم پیچ یعنی x3 باید از نوع عدد صحیح انتخاب گردد. تابع برازندگی و یا معیار و محدودیت‌های این مسئله نیز در رابطه۶ تشریح شده‌است. دراین میان قابل ذکر است که وزن فنر به طور کلی با حجم ماده‌ای که برای ساخت آن استفاده شده است، ارتباط دارد و حجم فنر به قطر سیم، تعداد دورها و طول فنر بستگی دارد. درجدول۱ الگوریتم کرم شب‌تاب با چندین الگوریتم بهینه‌سازی فراابتکاری دیگر از جمله الگوریتم مورچگان، رقابت استعماری، ازدحام ذرات،گرگ خاکستری و غیره مقایسه شده و همانطورکه مشاهده می‌شود الگوریتم کرم شب‌تاب پیشنهادی بهترین حل را با مقدار تابع هدف ۰/۰۱۲۶ ارائه می‌دهد.

اندازه ازدحام ۲۰ و حداکثر تعداد تکرار ۱۰۰ برای حل این مسأله بهینه سازی استفاده می‌شود.

همانطور که مشهود است الگوریتم DAPSO-GA از مرجع [۲۹] حل بهتری را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. درجدول۲ تمامی الگوریتم بهینه‌سازی فراابتکاری در کنار دیگر مقایسه شده و همانطورکه مشاهده می‌شود الگوریتم قورباغه پیشنهادی بهترین حل را با مقدار تابع هدف ۲۶۳/۸۰ ارائه می‌دهد.

$$x = [x_1\ x_2\ x_3] = [d\ \ D\ \ N]$$

$$, \ 0.25 \le x_2 \le 1.3,\ 2.0 \le 0.05 \le x_1 \le 2.0$$

$$x_3 \le 15.$$

$$J(x) = (x_3 + 2)x_2x_1^2$$

$$g_1(x): 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0.$$

$$g_2(x): \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(x): 1 - \frac{140.45x_1}{x_2^2x_3} \le 0.$$

$$g_4(x): \frac{x_1 + x_2}{1.5} - 1 \le 0$$

(۶)

جدول ۱- مساله طراحی فنرکششی فشاری[۲۹]

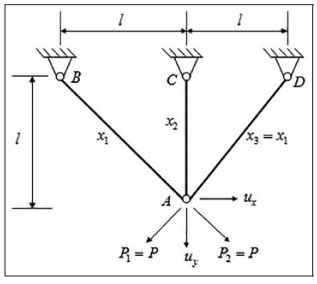
الگوریتم	x1	x2	x3	g1	g2	g3	g4	J
ICA	۰/۰۵۶۹	۰/۴۹۶۳	۶/۱۷۱۱	-۰/۱۰۵۳	-۰/۴۴۱۹	-۱/۵۸۴۲	-۰/۶۰۳۰	۰/۰۱۳۱
ABC	۰/۰۵۶۹	۰/۴۹۶۳	۶/۱۳۴۰	-۱/۰E-۴	-۱/۰E-۴	-۴/۲۶۶۴	-۰/۶۳۰۲	۰/۰۱۳۱
SELA	۰/۰۵۰۰	۰/۳۱۵۷	۱۴/۲۸۶۳	-۰/۲۱۷۸	-۰/۶۳۹۹	-۰/۵۷۳۱	-۰/۳۲۳۹	۰/۰۱۲۸
GWO	۰/۰۵۲۷	۰/۳۸۲۱	۹/۹۶۷۹	-۵/۲۴E-۴	-۷/۵۶E-۴	-۴/۰۲۳۶	-۵/۰۰۵۲	۰/۰۱۲۸
FA	۰/۰۵۱۰	۰/۳۳۹۴	۱۲/۳۸۵۸	-۱/۴۴E-۱۰	-۱/۵۱E-۱۱	-۴/۰۰۳۹	-۰/۵۵۱۸	**۰/۰۱۲۶**
IAPSO	۰/۰۵۱۶	۰/۳۵۶۶	۱۱/۲۹۴۱	-۱/۹۷E-۱۰	-۴/۶۴E-۱۰	-۴/۰۵۳۶	-۱/۰۹۱۶	۰/۰۱۲۶
MBA	۰/۰۵۱۶	۰/۳۵۵۹	۱۱/۳۴۴۶	-۰/۰۰۰۹	-۰/۱۳۴۴	-۴/۰۵۲۲	-۰/۷۲۸۲	۰/۰۱۲۶
DAPSO-GA	۰/۰۵۰۰	۰/۳۷۴۴	۸/۵۴۶۵	-۱/۲۵E-۸	-۱/۴۴E-۷	-۴/۸۶۰۷	-۰/۷۱۷۰	**۰/۰۰۹۸**

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

۲-۱-۱-۳. مسئله طراحی خرپا سه میله متقارن

شکل ۷ نمودار شماتیک یک سازه خرپایی متقارن سه میله‌ای را نشان می‌دهد. سازه خرپایی متقارن سه‌میله‌ای از فولاد تشکیل شده و تحت دو بار ثابت P1=P2=P قرار دارد. هدف مسئله، طراحی بهینه سازه خرپای سه‌میله‌ای و به‌حداقل رساندن حجم در معرض قیود تنش است. متغیرهای طراحی، سطح مقطع سه میله x1، x2 و x3 هستند.

تابع هزینه و محدودیت‌های این مسئله در رابطه (۷) آورده شده‌اند. الگوریتم‌های بخش۲ مقاله درکنار الگوریتم‌های مرجع[۲۹] با



شکل ۷. شماتیک یک سازه خرپایی متقارن سه میله ای[۲۹]

$$x = [\,x_1\ x_2] = [A_1\ A_2]$$

$$0 \le x_1, x_2 \le l$$

$$l = 1.m, P = 2.KN/cm^2, \sigma = 2.KN/cm^2$$

$$J(x) = \left(2\sqrt{2}x_1 + x_2\right) * l$$

$$g_1(x): \frac{\sqrt{2}\,x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0.$$

$$g_2(x): \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0.$$

$$g_3(x): \frac{1}{\sqrt{2}x_1 + 2x_1}P - \sigma \le 0.$$

(۷)

۳-۱-۱- مسأله طراحی کاهنده سرعت. درشکل۸ تصویر شماتیک یک کاهنده سرعت نشان داده شده است. طرح بهینه‌سازی طراحی کاهش‌دهنده سرعت، برای به‌حداقل رساندن وزن آن، با توجه به قیود استحکام مربوط به تنش خمشی دندانه‌های چرخ دنده و تنش سطحی و تنش‌ها و انحراف عرضی شفت‌ها است. متغیرهای طراحی این مسئله طراحی شامل عرض (b)، مدول دندانه (m)، تعداد دندانه‌ها در پینیون (z)، طول شفت اول و دوم بین یاتاقان‌های آنها (l1 و l2)، قطر شفت

اول (d1) و قطر شفت دوم (d2) است. این متغیرهای طراحی به ترتیب با x1، x2، x3، x4، x5، x_6 و x7 در

جدول ۲-مساله طراحی خرپای سه میله‌ای[۲۹]

الگوریتم	x_1	x_2	g_1	g_2	g_3	J
ACOR	۰/۸۰۸۳	۰/۳۵۵۳	-۱/۰E-۵	-۱/۵۲۵۸	-۱/۰۵۶۲	۲۶۴/۱۶
GWO	۰/۷۹۷۵	۰/۳۸۳۸	-۰/۶۰۱۲	-۱/۱۶۲۳	-۰/۶۲۳۱	۲۶۳/۹۵
FA	۰/۷۹۶۴	۰/۳۸۶۸	-۱/۳۱E-۹	-۶/۵۵۱۲	-۰/۵۱۲۳	۲۶۳/۹۳
ICA	۰/۷۸۳۳	۰/۴۲۳۸	-۰/۴۳۳۴	-۱/۴۳۳۹	-۰/۹۹۹۶	۲۶۳/۹۲
SFLA	۰/۷۸۵۶	۰/۴۱۷۱	-۰/۱۷۰۲	-۱/۳۷۹۶	-۰/۷۹۰۶	**۲۶۳/۸۰**
PSO-TVAC	۰/۷۸۸۷	۰/۴۰۸۱	-۴/۴۴E-۱۳	-۱/۴۶۴۲	-۰/۵۳۵۸	۲۶۳/۸۹
SC	۰/۷۸۸۶	۰/۴۰۸۴	-۸/۲۷E-۹	-۱/۴۶۳۹	-۰/۵۳۶۱	۲۶۳/۸۹
DAPSO-GA	۰/۷۸۸۶	۰/۴۰۸۲	-۲/۳۱E-۰۹	-۱/۴۶۴۱	-۰/۵۳۵۸	**۲۶۳/۸۹**

PSO-TVAC: Particle swarm optimization withtime-varying accelerating coefficients. ;SC :Society and civilization algorithm; DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm.

$$0.7 \le x_2 \le 0.8,\ 17 \le x_3 \le 28, 2.6 \le x_1 \le 3.6$$
$$5.0 \le x_7 \le 5.5, 7.3 \le x_4, x_5 \le 8.3, 2.9 \le x_6 \le 3.9$$

$$J(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

$$g_2(x): \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0. \quad g_1(x): \frac{27}{x_1x_2^2x_3} - 1 \le 0. \qquad (۸)$$

$$g_3(x): \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0. \quad g_4(x): \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0.$$

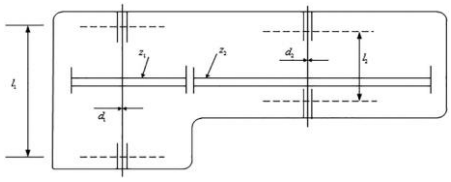
$$g_5(x): \frac{[(745\,x_4/x_2x_3)^2 + 16.9 \times 10^6]^{0.5}}{110x_6^3} - 1. \le 0$$

$$g_6(x): \frac{[(745\,x_6/x_2x_3)^2 + 157.5 \times 10^6]^{0.5}}{110x_7^3} - 1. \le 0.$$

معادلات تابع هدف و قیود نشان داده می‌شوند. متغیر طراحی x_3 (یعنی تعداد دندانه در پینیون) یک متغیر طراحی گسسته است و بقیه متغیرها پیوسته هستند. همچنین تابع برازندگی و یا معیار و محدودیت‌های این مسئله در رابطه۸ آورده شده‌اند.

الگوریتم‌های بخش۲ درکنار الگوریتم‌های مرجع[۲۹] با سایز ازدحام۴۰ وحداکثر تعداد تکرارهای۵۰۰ برای حل این مسئله بهینه-سازی استفاده می‌شود.

مقدار تابع هدف در تکرارهای اولیه به سرعت به حداقل نزدیک می‌شود



شکل ۸ .شماتیک یک کاهنده سرعت[۳۰]

(کمتر از ۵۰ تکرار)، که نرخ همگرایی بالای این الگوریتم را برای این مسأله طراحی گسسته مقید نشان می‌دهد. همانطور که نشان داده شده است، الگوریتم DAPSO-GA از مرجع [۲۹] حل مناسبی را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. درجدول۳ الگوریتم کرم شب-تاب نیز با الگوریتم بهینه‌سازی فراابتکاری دیگر مقایسه شده وهمانطورکه مشاهده می‌شود الگوریتم رقابت استعماری پیشنهادی بهترین حل را با مقدار تابع هدف ۲۹۹۰/۰۴ ارائه می‌دهد.

نسخه پیش انتشار

جدول ۳- مساله طراحی کاهنده سرعت[۲۹]

الگوریتم	x_1	x_2	x_3	x_4	x_5	x_6	x_7	J
GWO	۳/۵۱۹۸	۰/۷	۱۷	۷/۵۴۳۱	۸/۰۱۷۷	۳/۴۲۵۴	۵/۲۸۸۹	۳۰۳۲/۳۶
FA	۳/۵۳۴۴	۰/۷	۱۷	۷/۸۲۱۷	۸/۰۴۲۱	۳/۳۵۱۲	۵/۲۸۶۸	۳۰۲۰/۰۶
SFLA	۳/۵۰۰۰	۰/۷	۱۷	۷/۴۵۱۸	۷/۹۰۱۱	۳/۳۵۰۵	۵/۲۹۴۴	۳۰۰۴/۹۳
ACOR	۳/۵۰۳۰	۰/۷	۱۷	۷/۴۰۶۹	۷/۸۲۱۶	۳/۳۵۳۴	۵/۲۸۶۷	۳۰۰۴/۰۴
ICA	۳/۵۰۱۹	۰/۷	۱۷	۷/۳۰۰۲	۷/۷۱۵۷	۳/۳۵۰۲	۵/۲۸۶۷	**۲۹۹۰/۰۴**
IAPSO	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۳	۳/۳۵۰۲	۵/۲۸۶۶	۲۹۹۴/۴۷
MBA	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۷	۳/۳۵۰۲	۵/۲۸۶۶	۲۹۹۴/۴۸
DAPSO-GA	۳/۵۰۰۰	۰/۷	۱۷	۷/۳۰۰۰	۷/۷۱۵۳	۳/۳۵۰۲	۵/۲۸۶۶	**۲۹۹۴/۴۷**

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

۴-۱-۱-۳. مسأله طراحی مخزن تحت فشار

شکل۹ شماتیکی از یک مخزن تحت فشار را نشان می‌دهد. هدف طراحی به حداقل رساندن هزینه ساخت کل آن، از جمله هزینه های مواد، شکل‌دهی و جوش است. متغیرهای طراحی شامل ضخامت پوسته Ts، ضخامت سر Th، شعاع داخلی R و طول مقطع استوانه‌ای ظرف L، که در آن Ts و Th متغیرهای گسسته و مضرب صحیح ۰،۰۶۲۵ هستند، درحالی‌که R و L متغیرهای پیوسته هستند. این چهار متغیر طراحی به ترتیب با x1، x2، x3 و x4 در تابع هدف اعمال شده و معادلات قیود هم در رابطه ۹ ارائه شده است.

الگوریتم کرم شب‌تاب پیشنهادی با اندازه ازدحام ۲۵ و حداکثر تعداد تکرار ۵۰۰ عدد، برای حل این مسئله بهینه سازی استفاده می-شود. همانطور که مشهود است. الگوریتم MBA از مرجع [۲۹] حل بهتری را در مقایسه با سایر الگوریتم‌ها ارائه می‌دهند. درجدول۴ تمامی الگوریتم بهینه‌سازی فراابتکاری در کنار یکدیگر مقایسه شده و همانطورکه مشاهده می‌شود الگوریتم گرگ‌خاکستری پیشنهادی بهترین حل را با مقدار تابع هدف ۵۸۹۰/۷۵ پیدا می‌کند.

$$x = [\,x_1\ x_2\ x_3\ x_4\,] = [\,T_s\ T_h\ R\ L\,]$$

$$x_1 \geq 1 \times 0.0625,\ x_2 \leq 99 \times 0.0625,\ x_3 \geq 10.,\ x_4 \leq 200.$$

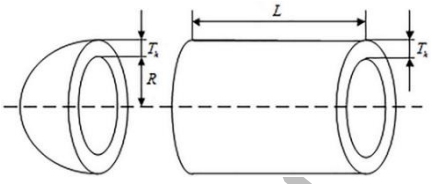
$$J(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.166x_1^2x_4 + 19.84x_1^2x_3 \quad (9)$$

$$g_1(x): -x_1 + 0.0193x_3 \leq 0.$$

$$g_2(x): -x_2 + 0.00954x_3 \leq 0$$

$$g_3(x): -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0.$$

$$g_4(x): x_4 - 240 \leq 0.$$



شکل ۹. شماتیک از یک مخزن تحت فشار[۳۰]

جدول ۴- مساله طراحی مخزن تحت فشار[۲۹]

الگوریتم	x_1	x_2	x_3	x_4	g_1	g_2	g_3	g_4	J
ACOR	۰/۸۴۶۲	۰/۴۱۷۸	۴۳/۸۹	۱۵۵/۵۶	۱/۰E-۳	۱/۰E-۳	۱/۰E-۳	-۸۴/۴۳	۶۰۰۴/۰۹
FA	۰/۸۰۰۷	۰/۳۹۷۱	۴۱/۳۷	۱۸۶/۰۴	-۱/۴E-۹	-۱/۵E-۴	-۰/۸E-۶	-۵۹/۵	۵۹۴۹/۰۶
ICA	۰/۷۸۴۲	۰/۳۸۵۴	۴۰/۵۳	۱۹۷/۰۷	-۰/۰۳۱۵	-۵/۲۶۹۷	-۲/۹۶E۴	-۱۳۰/۸۰	۵۹۰۳/۰۱
SFLA	۰/۷۸۰۶	۰/۳۸۴۸	۴۰/۴۳	۱۹۸/۵۶	-۰/۰۹۰۵	-۱/۷۱۷۷	-۳/۶E۴	-۱۱۱/۳۱	۵۸۹۱/۳۰
GWO	۰/۷۷۸۳	۰/۳۸۶۱	۴۰/۳۲	۱۹۹/۹۶	-۶/۲E-۱۵	-۱/۱E-۱۶	-۷۹/۹۹	-۴۴/۳	**۵۸۹۰/۷۵**
IAPSO	۰/۸۱۲۵	۰/۴۳۷۵	۴۲/۰۹	۱۷۶/۶۳	-۴/۰E-۱۳	-۳/۵E-۰۲	-۱/۳E-۰۷	-۶۳/۳	۶۰۵۹/۷۱
MBA	۰/۷۸۰۲	۰/۳۸۵۶	۴۰/۴۲	۱۹۸/۴۷	۰	۰	-۸۶/۳۶	-۴۱/۵	۵۸۸۹/۳۲
DAPSO-GA	۰/۸۱۲۵	۰/۴۳۷۵	۴۲/۰۹	۱۷۶/۶۳	-۴/۰E-۱۳	-۳/۵E-۰۲	-۱/۳E-۰۷	-۶۳/۳	**۶۰۵۹/۷۱**

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

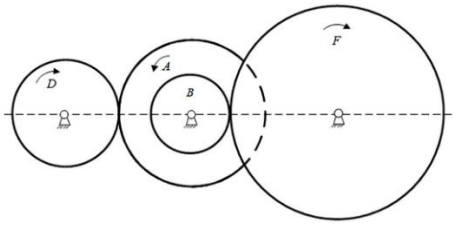
۳-۱-۲. مسائل بهینه‌سازی مهندسی مقیدبا متغیرهای گسسته[۲۹]

۳-۱-۲-۱. مساله طراحی مجموعه چرخ‌دنده‌ای.

شکل ۱۰ یک نمودار شماتیک از یک مجموعه دنده‌ای را نشان می‌دهد که از چهارچرخ‌دنده تشکیل شده است. طرح مسئله بهینه‌سازی طراحی مجموعه چرخ‌دنده‌ای، جهت به‌حداقل‌رساندن خطا بین نسبت دنده به‌دست‌آمده و نسبت دنده مورد نیاز، یعنی۱ برروی ۶/۹۳۱ بوده

$$x = [x_1\ x_2\ x_3\ x_4] = [N_1\ N_2\ N_3\ N_4]$$
$$12 \le x_i \le 60, i = 1,2,3,4 \qquad (۵)$$
$$J(x) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4}\right)^2$$

که برروی قیود که تعداد دندانه‌های چهار چرخ‌دنده است و فقط در محدوده مجاز متغیرهای طراحی است، اعمال می‌شود. این مسئله یک مسئله بهینه‌سازی گسسته است، زیرا همه متغیرهای طراحی اعداد صحیح هستند. تعداد دندانه‌های چرخ‌دنده‌های A، B، D و F (یعنی متغیرهای طراحی) در رابطه ۵ به‌ترتیب با x1، x2، x3 و x4 در تابع هدف، نشان داده شده است. تابع برازندگی و یا معیار و محدودیت‌های این مسئله نیز در رابطه۵ آورده شده‌اند.برای حل این مساله بهینه-سازی از الگوریتم کرم شب‌تاب گسسته با اندازه ازدحام ۱۰۰ و حداکثر تعداد تکرار ۵۰۰ استفاده شده است. همانطور که نشان داده شده است الگوریتم DAPSO-GA از مرجع [۳۰] حل مناسبی را در مقایسه با



شکل ۱۰. شماتیک از یک مجموعه چرخ دنده ای[۲۹]

سایر الگوریتم‌ها برای مساله طراحی مجموعه چرخ‌دنده‌ای ارائه می‌دهند. جدول۵ مقایسه راه‌حل‌های بهینه ارائه شده توسط روش‌های بخش۲ و الگوریتم‌های مرجع[۳۰]را ارائه می‌دهد. ملاحظه می‌شودکه الگوریتم رقابت استعماری پیشنهادی بهترین حل را با مقدار تابع هدف ۷/۷E-۳۴ ارائه می‌دهد.

جدول ۵- مساله طراحی مجموعه چرخ دنده ای[۲۹]

الگوریتم	x1	x2	x3	x4	J
FA	۵۹	۱۴	۱۴	۲۳	۲/۴۶E-۸
GWO	۵۹	۲۲	۱۲	۳۱	۳/۸۲E-۹
ACOR	۵۰	۲۱	۱۴	۴۳	۱/۸۸E-۱۲
SFLA	۵۷	۳۳	۱۴	۵۶	۳/۰۸E-۳۳
ICA	۴۵	۲۲	۱۳	۴۵	**۷/۷۰E-۳۴**
IAPSO	۴۳	۱۶	۱۹	۴۹	۲/۷۰E-۱۲
MBA	۴۳	۱۶	۱۹	۴۹	۲/۷۰E-۱۲
DAPSO-GA	۴۹	۱۶	۱۹	۴۳	**۲/۷E-۱۲**

IAPSO: improved adaptive particle swarm optimization; MBA: mine blast algorithm DAPSO-GA: dynamic adaptive particle swarm optimization and genetic algorithm

۴. نتایج

۴-۱. آزمایش حساسیت سنجی

بطورکلی معیار اعداد همگرایی در هر الگوریتم فراابتکاری بطور مجزا، نشان دهنده تعداد مراحل مورد نیاز برای رسیدن به بهینه مطلق است. هر چه عدد همگرایی کمتر باشد، الگوریتم بهتر و سریعتر به بهینه مطلق همگرا می‌شود. لذا در این مقاله به کمک آزمون حساسیت-سنجی روی پارامترهای مهم الگوریتم‌های فراابتکاری و بررسی حساسیت آنها نسبت به تغییر در هر یک از این پارامترها، بهترین تنظیمات پارامترها برای هر الگوریتم بهینه‌سازی جهت دستیابی به عدد همگرایی بهینه در حل تمام مسائل تعیین شده است.

بعنوان مثال در الگوریتم فراابتکاری قورباغه برای حل مسأله طراحی خرپای سه میله‌ای، پارامترهای بیشینه تعداد تکرار، جمعیت اولیه قورباغه ها، تعداد ممپلکس‌ها[1]، اندازه ممپلکس‌ها در آزمایش حساسیت‌سنجی مورد ارزیابی قرار گرفته و نتیجه نهایی حساسیت سنجی الگوریتم فراابتکاری قورباغه در شکل۱۱ در کنار بقیه الگوریتم-های فراابتکاری مشاهده می‌گردد.

۴-۲. نتایج همگرایی

شکل ۱۱ تاریخچه همگرایی تابع برازندگی مسأله طراحی خرپای سه میله‌ای برای پنج الگوریتم بهینه سازی فراابتکاری را نشان می‌دهد

[1] memeplex

و نتایج تأییدکننده موفقیت الگوریتم قورباغه برای مسأله طراحی خرپای سه میله‌ای است.

۳-۴. نمایش سرعت اجرا در حل الگوریتم‌ها.

در این بخش مطابق جدول۶ برای مسئله خرپای سه میله ای ، پنج الگوریتم فراابتکاری از نظر سرعت اجرا، با مقدار برازندگی مشخص و در ۱۰۰ تکرار موردارزیابی قرارگرفته است.

جدول ۶- نتیجه آزمون میانگین زمان اجرای الگوریتم‌ها

	نوع الگوریتم	زمان اجرا به ثانیه
۱	گرگ خاکستری	۱۸
۲	رقابت استعماری	۲۴
۳	کرم شبتاب	۲۵
۴	قورباغه	۴۰
۵	کلونی مورچگان	۴۲

در پایان نیز پیشنهاد می‌گردد روش‌های بررسی شده، با الگوریتم‌های کلاسیک، روش‌های مبتنی بر شبکه‌های عصبی و غیره مقایسه شده و ترکیب این روش‌ها نیز برای حل مسائل متنوع در دیگر حوزه‌ها و مسائل مهندسی مورد ارزیابی قرار گیرد.

۵. نتیجه گیری

در این مقاله پنج مسئله درحوزه مهندسی مکانیک به همراه قیود و نتایج مربوطه موردبررسی قرارگرفته است. الگوریتم‌های گرگ خاکستری، الگوریتم کلونی مورچگان، الگوریتم کرم شب‌تاب، رقابت استعماری و قورباغه در حل این مسائل مورد ارزیابی قرار گرفته‌اند و نتایج مربوطه شرح داده شده است. از نوآوری‌های این مقاله، ارضاء قیود مسائل است که در ابتدا به علت محدود بودن بازه جستجو و ایجاد همسایگی تصادفی در ناحیه‌ای که قیود و محدودیت‌ها را ارضاء می‌کند، انجام می‌شود. نکته دیگر استفاده از عامل تطبیقی پویا، در تمامی این الگوریتم ها است، بطوریکه با تغییر سرعت جستجو و تطبیق آن با شرایط محیطی، منجر به بهینه‌سازی بهتر می‌گردد.

نتایج این الگوریتم‌ها به همراه نتایج سایر محققین در جداول ۱ تا ۵ مقایسه شده‌اند.

بعنوان نمونه، نتایج حل مسأله طراحی خرپای سه میله‌ای توسط پنج الگوریتم بهینه سازی فراابتکاری نشان می‌دهند که الگوریتم قورباغه، با دقت ۰/۱۳۶درصد در این مسأله جواب دقیق‌تری دارد. درحقیقت برای مسائل مختلف باتوجه به گسسته یا پیوسته بودن مسئله، یک الگوریتم بهینه سازی مشخص، برای هر یک از مسائل پاسخ دقیقتری ارائه می‌دهد. این نتایج نشان می‌دهند که الگوریتم‌های فوق در مسائل مختلف می‌توانند همزمان دقت و سرعت را بهبود بخشیده و این از مظاهر هوش ازدحامی است.

۶. منابع و مراجع

[1] Guedria NB.Improved accelerated PSO algorithmform echanical engineering optimization problems.Appl Soft Comput (2016); 40: 455–467.

[2] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In Proceedings of ICNN'95-international conference on neural networks, vol. 4, pp. 1942-1948. ieee, (1995).

[3] Holland, John H. "Outline for a logical theory of adaptive systems." Journal of the ACM (JACM) 9, no. 3 (1962): 297-314.

[4] Bagley JD. The behavior of adaptive systems which employ genetic and correlation algorithms. Ann Arbor, MI: Dissertation Abstracts International, University of Michigan, (1967).

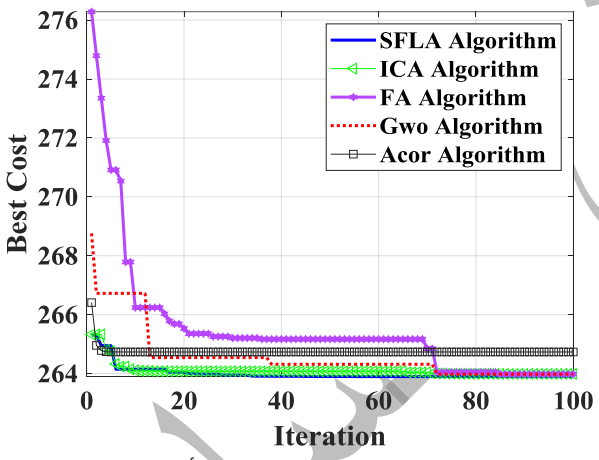
[5] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in engineering software 69 (2014): 46-61.

[6] Li, Yi, Wenwen Sun, and Shaopeng Guan. "A firefly inspired controller placement algorithm in software defined network." In 2019 IEEE 2nd international conference on computer and communication engineering technology (CCET), pp. 254-258. IEEE, (2019).

[7] Dorigo M, Birattari M and Stutzle T. Ant colony optimization artificial ants as a computational intelligence technique.IEEE Comput Intell M (2006); 1: 28–39.

[8] Karaboga D and Basturk B. Artificial bee colony (ABC)optimization algorithm for solving constrained optimization. In: Melin P, Castillo O, Aguilar LT, et al. (eds)Foundations of fuzzy logic and soft computing.IFSA 2007(Lecture Notes in Computer Science, vol. 4529).Berlin:Springer,(2007),pp.789–798.

[9] Ahandani, Morteza Alinia, Reza Banimahd, and Naser Pourqorban Shrjoposht. "Solving the parameter identification problem using shuffled frog leaping with opposition-based initialization." In



شکل ۱۱. تاریخچه همگرایی تابع هزینه مسأله طراحی خرپای سه میله ای توسط پنج الگوریتم بهینه سازی فراابتکاری

2011 1st International eConference on Computer and Knowledge Engineering (ICCKE), pp. 49-53. IEEE, (2011).

[10] Xing, Bo, Wen-Jing Gao, Bo Xing, and Wen-Jing Gao. "Imperialist competitive algorithm." Innovative computational intelligence: A rough guide to 134 clever algorithms (2014): 203-209.

[11] Aala Kalananda, Vamsi Krishna Reddy, and Venkata Lakshmi Narayana Komanapalli. "A competitive learning-based Grey wolf Optimizer for engineering problems and its application to multi-layer perceptron training." Multimedia Tools and Applications (2023): 1-59.

[12] Lunardi, Willian Tessaro, Holger Voos, and Luiz Henrique Cherri. "An effective hybrid imperialist competitive algorithm and tabu search for an extended flexible job shop scheduling problem." In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 204-211. (2019).

[13] Rajabioun, Ramin, Esmaeil Atashpaz-Gargari, and Caro Lucas. "Colonial competitive algorithm as a tool for Nash equilibrium point achievement." In Computational Science and Its Applications–ICCSA 2008: International Conference, Perugia, Italy, June 30–July 3,

2008, Proceedings, Part II 8, pp. 680-695. Springer Berlin Heidelberg, (2008).

[14] Eusuff, Muzaffar, Kevin Lansey, and Fayzul Pasha. "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization." Engineering optimization 38, no. 2 (2006): 129-154.

[15] Maaroof, Bestan B., Tarik A. Rashid, Jaza M. Abdulla, Bryar A. Hassan, Abeer Alsadoon, Mokhtar Mohammadi, Mohammad Khishe, and Seyedali Mirjalili. "Current studies and applications of shuffled frog leaping algorithm: a review." Archives of Computational Methods in Engineering 29, no. 5 (2022): 3459-3474.

[16] Eusuff, Muzaffar, Kevin Lansey, and Fayzul Pasha. "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization." Engineering optimization 38, no. 2 (2006): 129-154. pp.94–97. New York: IEEE.

[17] Yang, Xin-She, and Xingshi He. "Firefly algorithm: recent advances and applications." International journal of swarm intelligence 1, no. 1 (2013): 36-50.

[18] Johari, Nur Farahlina, Azlan Mohd Zain, Mustaffa H. Noorfa, and Amirmudin Udin. "Firefly algorithm for optimization problem." Applied Mechanics and Materials 421 (2013): 512-517.

[19] Yang, Xin-She. "Firefly algorithm, stochastic test functions and design optimisation." International journal of bio-inspired computation 2, no. 2 (2010): 78-84.

[20] Farahani, Sh M., Azam A. Abshouri, Babak Nasiri, and MR2011 Meybodi. "A Gaussian firefly algorithm." International Journal of Machine Learning and Computing 1, no. 5 (2011): 448.

[21] Kumar, Vijay, and Dinesh Kumar. "A systematic review on firefly algorithm: past, present, and future." Archives of Computational Methods in Engineering 28 (2021): 3269-3291.

[22] Yang, Xin-She. "Firefly algorithms for multimodal optimization." International symposium on stochastic algorithms. Berlin, Heidelberg: Springer Berlin Heidelberg, (2009).

[23] Alizadeh, Mohammad Hosein, and Alireza Toloei. "Designing Pitch Angle Compensator for an UAV and Robustification it with Bee Colony Optimization Algorithm." Technology in Aerospace Engineering (2023): 1-13.

[24] Attaran, Behrooz, Milad Zarchi, and A. R. Toloei. "Numerical survey of vibrational model for third aircraft based on HR suspension system actuator using two bee algorithm objective functions." International Journal of Engineering 30, no. 6 (2017): 887-894.

[25] Toloei, Ali Reza, Milad Zarchi, and Behrooz Attaran. "Application of active suspension system to reduce aircraft vibration using pid technique and bees algorithm." International Journal of Computer Applications 98, no. 6 (2014).

[26] Toloei, Ali Reza, Milad Zarchi, and Behrooz Attaran. "Vibration control of aircraft semi-active suspension system using PID-bees technique." International Journal of Computer Applications 975 (2014): 8887.

[27] Ameri Mould, and Behrouzian Nejad Ebrahim. "Presenting a clustering method based on colonial competition algorithm to increase network lifetime in wireless sensor networks." (۲۰۰۵). (InPersian)

[۲۸] Ghafari Moghadam, Hashemi Tabar, and Moradi. "Water DemandManagement Using Dynamic Firefly Optimization Algorithm: A Case Study of Hirmand River." Scientific Quarterly Journal of Water Resources Engineering ۱۳,no. ۴۶ (۲۰۲۰): ۸۵-۹۸(InPersian)

[۲۹] Zhu, Hao, Yumei Hu, and Weidong Zhu. "A dynamic adaptive particle swarm optimization and genetic algorithm for different constrained engineering design optimization problems." Advances in Mechanical Engineering 11, no. 3 (2019): 1687814018824930.

[۳۰] Okonta, C. I., A. H. Kemp, R. O. Edopkia, G. C. Monyei, and E. D. Okelue. "A heuristic based ant colony optimization algorithm for energy efficient smart homes." In Proc. 5th Int. Conf. Exhib. Clean Energy, pp. 1-12. (2016).