



افزایش کارایی ترکیب سرویس‌های ابری مبتنی

بر الگوریتم کرم شب‌تاب بهبودیافته نوین

بهنام فرناد^(۱) کامبیز مجیدزاده^{(۲)*} محمد مصدری^(۳) امین بابازاده سنگر^(۴)

(۱) گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

(۲) گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران*

(۳) گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

(۴) گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

(تاریخ دریافت: ۱۴۰۲/۰۹/۲۶ تاریخ پذیرش: ۱۴۰۳/۰۳/۱۷)

چکیده

با افزایش تکنولوژی‌های از راه دور در زندگی انسان‌ها استفاده از سرویس‌های ابری پیچیده‌تر شده و ترکیبی از چندین سرویس را دربر می‌گیرد. درخواست سرویس، سرعت پاسخ‌دهی و تولید ترکیبی از سرویس‌ها باید به صورتی باشد که بیشترین کیفیت سرویس‌دهی را در کمترین زمان ارائه دهد. روش مطرح شده توانسته است، کیفیت سرویس بالا را در ترکیب سرویس‌ها با کمترین زمان نسبت به سایر روش‌های پایه فرا ابتکاری بدست آورد. در این روش الگوریتم کرم شب‌تاب با علوم زیستی و رفتاری جدید بروزرسانی شده است. رفتارهای جدید اضافه شده شکارچی و حرکت به سمت جمع، سبب ایجاد تعادل بین قابلیت اکتشاف و بهره‌برداری الگوریتم مطرح شده می‌شود. مشکل اکثر روش‌ها، بدست آوردن کیفیت بالای ترکیب سرویس در مدت زمان کمتر می‌باشد که در این الگوریتم با اضافه کردن رفتارهای جدید برطرف شده است. نمودارهای همگرایی برحسب زمان و نتایج آماری بدون پارامتر برتری این روش را نسبت به سایر روش‌ها فرا ابتکاری نشان می‌دهد. این روش داده‌های استاندارد QWS و توابع استاندارد را در مقایسه با الگوریتم‌های مطرح در کمترین زمان با بهترین جواب حل می‌کند.

کلمات کلیدی: بهینه‌سازی، الگوریتم کرم شب‌تاب، ترکیب سرویس‌های ابری، فرا ابتکاری.

*عهده‌دار مکاتبات:

کامبیز مجیدزاده

نشانی: گروه مهندسی کامپیوتر، واحد ارومیه، دانشگاه آزاد اسلامی، ارومیه، ایران

پست الکترونیکی: Kambiz.majidzadeh@iau.ac.ir

رایانش ابری یک روش برای ارائه خدمات فناوری اطلاعات به مشتریان در قالب سرویس به همراه انعطاف‌پذیری و مقیاس‌پذیری می‌باشد که مشتری‌ها بدون درگیری با پیچیدگی‌های سیستمی، قابلیت استفاده از منابع، پلتفرم و یا نرم‌افزاری را در صورت درخواست آن دارند. امروزه سرویس‌های مختلف ابری با قیمت‌ها و قابلیت‌های مختلف ارائه می‌شود تا نیازمندی‌های کاربران را برطرف کند زیرا که یک سرویس به تنهایی قابلیت ایجاد کیفیت مناسب را ندارد و این ترکیب سرویس‌های مختلف با قابلیت‌های متفاوت است که می‌تواند مشتریان را راضی سازد.

سرویس‌های موازی عملیات یکسانی را انجام می‌دهند و هر سرویس انتزاعی (کار) دارای چندین سرویس موازی است. چالش اصلی که اینجا مطرح می‌شود، انتخاب سرویس مناسب از میان سرویس‌های موازی برای هر سرویس انتزاعی است. انتخاب سرویس موازی بر پایه میزان پارامترهای کیفیت سرویس (QOS) می‌باشد، زیرا همه سرویس‌های موازی یک عملیات مشخص دارند و تفاوت QOS برای این سرویس‌های موازی تعیین‌کننده برتری سرویس است [1].

از آنجا که انتخاب سرویس‌های موازی می‌تواند در روش‌های مختلف صورت بگیرد و این انتخاب‌ها باعث ایجاد جواب‌های به تعداد نامایی شود که نیازمند پردازش به زمان نامایی است. تعداد ترکیب سرویس‌های ابری که می‌توان بدست آورد به اندازه m^n می‌باشد که در اینجا m تعداد سرویس‌های انتزاعی (کار) و n تعداد سرویس‌های کاندید (موازی) است. از آنجا که مسئله ترکیب سرویس‌های ابری یک مسئله نوع سخت ($NP-Hard$) می‌باشد نیازمند آن هستیم که روشی پیدا کنیم که بتواند بهترین حالت را برای تمام ویژگی‌های کاربردی و غیرکاربردی سرویس در نظر بگیرد که همین امر سبب می‌شود به یک مسئله بهینه‌سازی تبدیل شود.

در میان روش‌های بهینه‌سازی می‌توان به یکی از مجموعه روش‌های بسیار مناسب برای حل مسائل سخت اشاره کرد که آن هم روش‌های فرا ابتکاری [2] می‌باشد. این روش‌ها می‌تواند مسائل سخت را در پیچیدگی زمانی چند جمله‌ای حل کرده و دیگر نیازمند به روش‌های با پیچیدگی زمانی نامایی نباشیم. جوابی که الگوریتم‌های فرا ابتکاری بدست می‌آورند یا دقیق هستند و یا نزدیک به جواب درست می‌باشند. بیشتر روش‌های فرا ابتکاری بر پایه جمعیت بوده و اکثراً از هوش جمعی برای پیدا کردن جواب اصلی مسئله استفاده می‌کنند. روش‌های فرا ابتکاری بیشتر از طبیعت الگو گرفته شده‌اند و از شبیه‌سازی رفتارهای موجودات مختلف بهره می‌برند.

در این مقاله، الگوریتم بهینه‌سازی کرم شب‌تاب (FA) [3] که یکی از پرکاربردترین الگوریتم‌های فرا ابتکاری بر گرفته از طبیعت می‌باشد را مورد تغییر و تحول قرار می‌دهیم. نوآوری که در الگوریتم مطرح شده اعمال شده است مربوط به بررسی رفتارهای جدید کرم شب‌تاب می‌باشد. با بررسی این رفتارهای جدید و الگو گرفتن از آنها، ساختار الگوریتم با اضافه کردن بخش‌های جدید فاز شکارچی و حرکت به سمت جمع تغییر داده شده است. این تغییرات اعمال شده سبب شده است که الگوریتم از لحاظ قابلیت اکتشاف ($exploration$) و بهره‌برداری ($exploitation$) به تعادل برسد و محیط بیشتری را با دقت بهتر

جستجو کند و کمتر در دام بهینه‌های محلی بیافتد. بررسی‌ها نشان می‌دهد که حرکت کرم شب‌تاب‌ها به دو علت اساسی است، اولاً برای جفت‌گیری که کرم شب‌تاب ماده شروع به نوردی به کرم شب‌تاب نری می‌کند که نزدیکتر است و شدت نور بیشتری دارد و دوماً بعضی وقت‌ها کرم شب‌تاب ماده از نر برای تغذیه کردن استفاده می‌کند (رفتار شکارچی) [4]. کرم شب‌تاب‌ها در بررسی‌های قبلی بدین صورت بودن که فقط به سمت کرم شب‌تاب پور نور حرکت می‌کردند، ولی یافته‌های جدید نشان می‌دهد که حرکت این کرم‌ها همچنین تحت تاثیر حرکت به سمت جمع نیز می‌باشد، یعنی کرم شب‌تاب علاوه بر اینکه تمایل دارد به سمت کرم شب‌تاب پرنور حرکت کند تمایل در حضور جمعی و همه کرم شب‌تاب‌ها دارد [5] که برای شبیه‌سازی این حرکت نیازمند این هستیم که علاوه بر اینکه کرم شب‌تاب تحت تاثیر پر نورترین محلی باشد تحت تاثیر بهینه سراسری نیز باشد که سبب کشیده شدن آن به سمت جمع و دسته‌ها در کرم شب‌تاب‌ها می‌شود. پیاده‌سازی رفتار شکارچی سبب می‌شود که در جمعیت کرم شب‌تاب‌ها برخی از جواب‌ها حذف شده و با جواب‌های جدید جایگزین شود و این عمل سبب دگرگونی در جمعیت جواب‌ها می‌شود و قابلیت اکتشاف را افزایش می‌دهد و از سمت دیگر با اضافه کردن رفتار حرکت به سمت جمع، کرم شب‌تاب تحت تاثیر جمع نیز می‌باشد. این امر سبب می‌شد که جواب‌ها بیشتر در ناحیه اطراف بهترین جواب بررسی شود و قابلیت بهره‌برداری را افزایش دهد. روش مطرح شده را در سه زمینه مختلف بهینه‌سازی توابع استاندارد، ترکیب سرویس‌های ابری داده‌های تصادفی و مجموعه داده‌های *QWS* مورد ارزیابی قرار داده می‌شود. آزمون‌های بدون پارامتر و بررسی‌های نمودارهای همگرایی برحسب زمان نشان می‌دهد که روش مطرح شده نسبت به روش پایه الگوریتم کرم شب‌تاب و الگوریتم‌های معروف دیگر فرا ابتکاری بهتر عمل می‌کند. این روش داده‌های استاندارد *QWS* را در مقایسه با الگوریتم‌های مطرح فرا ابتکاری در کمتر از ۲۰ ثانیه با بهترین مقدار کیفیت سرویس *QOS* ترکیب سرویس می‌کند. روش مطرح شده نیز عملکرد بهتری از *FA* در سرعت دست رسی به کیفیت بالای ترکیب سرویس‌های ابری دارد و می‌توان آن را به جای الگوریتم کرم شب‌تاب استفاده کرد. جدول (۱) مقایسه تغییرات ساختاری روش مطرح شده با روش قبلی الگوریتم *FA* را نشان می‌دهد.

بخش‌های این مقاله به این صورت می‌باشد، ابتدا در بخش دوم به مرور ادبیات و کارهای گذشته پرداخته شده است، در بخش سوم به معرفی مسئله ترکیب سرویس‌های ابری و پیاده‌سازی آن پرداخته شده است. در بخش چهارم، روش پیشنهادی و نحوه پیاده‌سازی آن بر روی ترکیب سرویس‌های ابری می‌پردازیم و در بخش پنجم ارزیابی کاملی از روش مطرح شده ارائه شده است و روش مطرح شده برای ترکیب سرویس ابری در داده‌های مطرح شده در پایگاه داده *QWS* استفاده می‌شود و نهایتاً در بخش آخر، نتیجه کلی از عملکرد این روش توضیح داده می‌شود.

جدول ۱: مقایسه روش مطرح شده با روش قبلی الگوریتم کرم شب تاب

ویژگی‌ها	الگوریتم کرم شب تاب قدیم (<i>FA</i>)	الگوریتم کرم شب تاب مطرح شده (<i>NFA</i>)
طبیعت الگو گرفته شده	براساس رفتار کرم شب تاب برای جا به جایی	براساس رفتار کرم شب تاب برای جابه جایی و تغذیه (شکارچی)
قوانین روشنایی	کرم شب تاب به سمت کرم شب تاب پر نور جذب می شود.	کرم شب تاب علاوه بر حرکت به سمت پر نور تر در شعاع محلی خود به سمت پر نورترین کرم شب تاب کلی (بهترین جواب) نیز حرکت می کند.
محدودیت‌ها و مزیت‌ها	نبود قابلیت اکتشاف زیاد و سرعت پایین همگرایی	افزایش قابلیت اکتشاف با اضافه کردن رفتار تغذیه که سبب اضافه شدن کرم شب تاب های جدید به جمعیت شده و تغییر سرعت همگرایی با تحت تاثیر قرار دادن حرکت، توسط بردار بهترین کرم شب تاب (حرکت به سمت جمع)

۲- مرور ادبیات و روش های پیشین

اغلب روش های فرا ابتکاری برگرفته از طبیعت با بررسی کلی فضای جستجو و پخش کردن جواب های تصادفی در محیط شروع به بازرسی می کنند و با تولید جواب های تصادفی از نقاط پخش شده در محیط و استفاده از هوش شبیه سازی شده از رفتارهای موجودات، اقدام به پیدا کردن بهترین جواب می کنند. از میان روش های فرا ابتکاری برگرفته از طبیعت می توان به الگوریتم زنبور عسل مصنوعی (*ABC*) [6]، الگوریتم جستجوی ازدحام ذرات (*PSO*) [7]، الگوریتم گرگ خاکستری (*GWO*) [8]، الگوریتم بهینه سازی وال (*WOA*) [9]، الگوریتم جستجوی آخوندک (*MSA*) [10]، بهینه سازی فوک آبی (*HSWOA*) [11]، بهینه سازی گراز دریایی (*WO*) [12]، بهینه ساز مار (*SO*) [13]، بهینه سازی ضد ویروس کرونا (*ACVO*) [14]، بهینه ساز فاجعه چرنوبیل (*CDO*) [15]، بهینه ساز دره انرژی (*EVO*) [16] و بهینه ساز روباه های پرنده (*FFO*) [17] اشاره کرد. از آنجا که مسئله ترکیب سرویس ابری، دارای جواب های محلی بسیاری خواهد بود، پس نیازمند آن هستیم که خاصیت اکتشاف را برای الگوریتم فرا ابتکاری مد نظر افزایش دهیم تا بتوانیم این روش را برای ترکیب سرویس های ابری استفاده کنیم. خاصیت اکتشاف (*exploration*) سبب می شود ناحیه جواب های بسیاری را در جستجو داشته باشیم که همین امر از افتادن در تله جواب های بهینه محلی جلوگیری کرده و سبب می شود به سمت بهینه سراسری حرکت کنیم.

در میان روش های فرا ابتکاری که برای بهینه سازی سرویس های ابری استفاده شده است، تاکنون روش های بسیاری مطرح شده است. محققان اخیرا با تغییر الگوریتم علف هرز (*IWO*) [18] روشی را برای حل مسئله ترکیب سرویس ابری مطرح کرده اند. این روش بر روی مجموعه داده های از دنیای واقعی مورد ارزیابی قرار گرفته و توانسته است نسبت به روش های دیگر بهتر عمل کند [19]. روشی دیگری نیز برگرفته از الگوریتم ژنتیک و با استفاده از تکنیک های داده کاوی مطرح شده است که آنها توانسته اند در بهینه سازی ترکیب سرویس ها سرعت عملیات را افزایش دهند [20]. اخیرا روشی نیز با استفاده از ترکیب الگوریتم ژنتیک و ازدحام ذرات استفاده شده است که این روش بر روی داده ها *QWS* ارزیابی شده و توانسته است از الگوریتم های پایه خود یعنی ژنتیک و ازدحام ذرات بهتر عمل کند [21]. روشی جدید از الگوریتم ازدحام ذرات برای ترکیب سرویس های ابری نیز مطرح شده است که این روش از الگوریتم تطبیق الگوی سرویس استفاده می کند که نتایج حاصل از این الگوریتم وارد الگوریتم

از دحام ذرات بهبود یافته می‌شود. روشی که این محققان مطرح کردند توانسته است از لحاظ بهینگی، پیچیدگی زمانی و همگرایی نسبت به روش‌های مطرح دیگر عمل کند [22].

از الگوریتم‌های که اخیراً با تغییر دادن و بهینه کردن الگوریتم کرم شب‌تاب پیاده‌سازی شده‌اند، می‌توان به روش‌های *GOQRFA* [24]، *MFA* [25] و *MFA-ANN* [26] اشاره کرد. در روش *GOQRFA* با استفاده از عملگرهای الگوریتم ژنتیک و روش یادگیری شب‌بازتابی اقدام به تغییر الگوریتم کرم شب‌تاب کرده‌اند که برای زمان بندی گردش کار در محیط ابری استفاده شده است. روش *MFA* یک روش ترکیبی از الگوریتم کرم شب‌تاب و رگرسیون برداری است و روش *MFA-ANN*، روشی است که در آن با تنظیم دقیق‌تر پارامترها و به صورت تطبیقی اقدام به بهبود الگوریتم کرم شب‌تاب کرده‌اند. روش مطرح شده *NFA* یک روش الگو گرفته شده از طبیعت است و مانند روش‌های بررسی شده، یک روش ترکیبی و یا تغییر یافته برحسب نیاز نیست بلکه این روش به طور کامل از رفتارهای جدید کرم شب‌تاب که اخیراً مطرح شده است الگو گرفته شده است و در واقع این روش مطرح شده حالت کامل الگوریتم کرم شب‌تاب می‌باشد چون همه رفتارهای کرم شب‌تاب شبیه‌سازی و به آن اضافه شده است.

۲-۱- مبانی نظری

۲-۱-۱- الگوریتم کرم شب‌تاب

الگوریتم کرم شب‌تاب اولین بار توسط *X.S Yang* در سال ۲۰۰۷ معرفی شد. این الگوریتم از رفتار کرم شب‌تاب‌ها در حرکت و جذب شدن به سمت نور شبیه‌سازی شده‌اند. حشرات چشمک زن در شب همان کرم شب‌تاب‌ها هستند که هنگام چشمک زدن دور هم جمع می‌شوند. هر کرم شب‌تاب یک ریتم منظم از نور را از خود ساطع می‌کنند که شدت جذابیت نور I دارد و این مقدار با بیشتر شدن فاصله r از منبع نور کاسته می‌شود. پس هر کرم شب‌تاب به کرم شب‌تابی جذب می‌شود که به آن نزدیک‌تر و پرنورتر است. شدت نور در الگوریتم کرم شب‌تاب، ارزش تابع برازش را نشان می‌دهد به این معنا که هر چقدر یک کرم شب‌تاب پرنورتر باشد آن کرم شب‌تاب نسبت به بقیه، برازنده‌تر و بهینه‌تر می‌باشد. شدت جذابیت نور یک کرم شب‌تاب در الگوریتم کرم شب‌تاب با I مشخص می‌شود. برای بررسی بیشتر جزئیات این الگوریتم به مرجع [26] مراجعه کنید.

الگوریتم کرم شب‌تاب به صورت زیر می‌باشد.



(ب)



(الف)

شکل ۱: الف) تغذیه کرم شب تاب توسط کرم شب تاب شکارچی، ب) تمایل زندگی کرم شب تاب در میان جمعیت،

شبه کد الگوریتم کرم شب تاب (FA):

- 1: Generate the initial population randomly.
- 2: Calculate the fitness of initial population based on light intensity of fireflies.
- 3: **While** (criterion)
- 4: **for** $i = 1, 2, \dots, n$ **do** (n firefly)
- 5: **for** $j = 1, 2, \dots, n$ **do**.
- 6: Calculate β .
- 7: Distance between two fireflies
- 8: **If** ($I(i) < I(j)$) **then**
- 9: firefly i is moved towards firefly j
- 10: Determine new solutions..
- 11: **else**
- 12: Firefly i is moved randomly towards j .
- 13: **end if**
- 14: **end for j**
- 15: **end for i**
- 16: **end while**
- 17: Sort the fireflies according to light intensity values of the new solution.

۲-۱-۲- رفتارهای جدید کرم شب تاب

الگوریتم کرم شب تاب تاکنون دچار تغییراتی زیادی شده و مورد بررسی قرار گرفته است، ولی هیچ موقع یافته‌های زیستی جدید در مورد این موجود زیبا و هوشمند بررسی نشده است، یافته‌هایی که می‌تواند گام جدید در بهینه‌سازی برای این الگوریتم باشد. با بررسی یافته‌های جدید در این الگوریتم تغییراتی در تابع بروزرسانی این الگوریتم داده و حالت‌های جدیدی را می‌افزاییم، ابتدا به بررسی رفتارهای می‌پردازیم که قبلاً در کرم شب تاب مطرح نشده بود.

۲-۱-۲-۱- حرکت به سمت جمع

کرم شب‌تاب‌ها در بررسی‌های قبلی بدین صورت بودن که فقط به سمت کرم شب‌تاب پور نور حرکت می‌کنند، ولی یافته‌های جدید نشان می‌دهد که حرکت این کرم‌ها همچنین تحت تاثیر حرکت به سمت جمع نیز می‌باشد، یعنی کرم شب‌تاب علاوه بر اینکه تمایل دارد به سمت کرم شب‌تاب پرنور حرکت کند تمایل در حضور جمعی و همه کرم شب‌تاب‌ها دارد (شکل (۱)) که برای شبیه‌سازی این حرکت نیازمند این هستیم که علاوه بر اینکه کرم شب‌تاب تحت تاثیر پرنورترین محلی باشد تحت تاثیر بهینه سراسری نیز باشد که سبب کشیده شدن آن به سمت جمع و دسته‌ها در کرم شب‌تاب‌ها می‌شود [5].

بررسی‌ها نشان می‌دهد که حرکت کرم شب‌تاب‌ها به دو علت اساسی است، اولاً برای جفت‌گیری که کرم شب‌تاب ماده شروع به نوردی به کرم شب‌تاب نری می‌کند که نزدیک تر است و شدت نور بیشتری دارد و دوماً بعضی وقت‌ها کرم شب‌تاب ماده از نر برای تغذیه کردن استفاده می‌کند.

۲-۲-۱-۲- تغذیه کرم شب‌تاب نر

نوردی کرم شب‌تاب‌ها همیشه برای جفت‌گیری نیست، بعضی وقت‌ها کرم شب‌تاب‌های شکارچی این عمل را تقلید کرده و سبب شکار کرم شب‌تاب از نوع دیگر می‌شود (شکل (۱)). برای شبیه‌سازی همچنین عملی در کرم شب‌تاب‌ها که یک رفتار پارازیتسم [27] در اکوسیستم هست (در رفتار پارازیتسم، از میان دو موجود که با یکدیگر در تعامل می‌باشند یک طرف همیشه رفتار انگلی یا شکارچی داشته و سبب بیماری و یا از بین رفتن طرف دیگر می‌شود) از یک متغیر تصادفی تصمیم‌گیری استفاده می‌کنیم. این متغیر مشخص می‌کند که چه مقدار احتمال دارد کرم شب‌تابی که نوردی می‌کند، شکارچی باشد. در صورتی که این کرم شکارچی باشد آرایه شکارچی ($X_{parasitism}$) را از کرم شب‌تابی که قرار است خورده شود می‌سازیم تا بررسی کنیم که آیا خورده خواهد شد یا نه در صورتی که آرایه بهتر نسبت به کرم شب‌تاب باشد پس شکارچی موفق بوده و کرم شب‌تاب حذف شده و آرایه شکارچی جایگزین کرم شب‌تاب می‌شود در غیر این صورت کرم شب‌تاب موفق به فرار می‌شود و این عمل اتفاق نمی‌افتد [4].

به طور کلی حرکت کرم شب‌تاب تحت تاثیر نورهای جمعی و کرم شب‌تاب پرنور می‌باشد، کرم شب‌تاب X_i در محیط جستجو دوبعدی تحت تاثیر نور کرم شب‌تاب پرنور محلی X_j قرار گرفته و به سمت آن حرکت می‌کند، در عین حال به سمت بهترین جواب $BestSol$ که سبب حرکت به سمت جمع کرم شب‌تاب‌ها می‌شود نیز تمایل دارد، پس مکان حرکت کرم شب‌تاب توسط بردار برآیند $newsol$ مشخص می‌شود و در نواحی آن بردار سمت کرم شب‌تاب پرنور می‌رود.

۳- مسئله ترکیب سرویس‌های ابری و پیاده‌سازی آن

در مسئله ترکیب سرویس‌های ابری به تعداد m کار ($Task$) جدا از هم وجود دارد که این کارها را به صورت

$\{Task_1, Task_2, Task_3, \dots, Task_m\}$ نمایش می‌دهیم. هر کار به شماره i ($Task_i$) می‌تواند توسط سرویس انتزاعی S_i انجام شود که از گروهی از سرویس‌های با کارایی یکسان ولی QoS متفاوت تشکیل شده اند که n_i اندازه تعداد سرویس‌های است که به S_i تعلق می‌گیرد و به صورت عبارت $\{ws_{i,1}, ws_{i,2}, ws_{i,3}, \dots, ws_{i,n_i}\}$ نمایش داده می‌شود. در طول ترکیب سرویس‌های ابری $ws_{i,j}$ یک سرویس کاندید است که موجودیت یک سرویس مشخص را برای کار مشخص نشان می‌دهد. کارایی یک سرویس به تنهایی بسیار ضعیف است، برای یکسری درخواست‌های پیچیده نیازمند ترکیب سرویس‌ها می‌باشیم که آن ترکیب به صورت $Q(CS) = \{Q_1, Q_2, Q_3, \dots, Q_r\}$ نمایش داده می‌شود. که در اینجا Q_k نشان دهنده میزان تجمیع بعد k ام از CS می‌باشد که توسط مقادیر صفات هر کدام از سرویس‌ها تجمیع شده است، در اینجا r تعداد صفات می‌باشد (CS همان سرویس موازی مد نظر است). کارایی در خدمات ابری شامل تعادل انعطاف پذیری، قابلیت اطمینان، عملکرد و مقرون به صرفه بودن است. ترکیب سرویس به کسب و کارها اجازه می‌دهد تا با ترکیب سرویس‌های ابری مختلف برای برآوردن نیازهای خاص خود، گردش کار سفارشی‌سازی شده ایجاد کنند. در جدول (۲) نمادهای استفاده شده، توصیف داده شده اند.

جدول ۲: جدول نمادهای مسئله ترکیب سرویس ابری

نماد	توضیح	نماد	توضیح
m	تعداد کار ($Task$)	$Task_i$	کار i ام
S_i	سرویس انتزاعی i ام	n_i	تعداد سرویس‌های موازی سرویس دهند انتزاعی i ام
$ws_{i,j}$	یک سرویس کاندید برای کار مشخص	Q_k	میزان تجمیع بعد k ام
r	تعداد صفات	CS	سرویس موازی مد نظر
w_k	وزن صفت k ام	q	مشخصه کیفیت سرویس

مدل QoS در سرویس‌ها، یکی از شاخص‌های مهم ارزیابی کیفیت سرویس می‌باشد و از چندین بعد تشکیل می‌شود تا بتوانیم سرویس را از جنبه‌های مختلف ارزیابی کنیم مانند: زمان پاسخ دهی ($response\ time$)، هزینه ($price$)، تکرار ($reputation$) و

در این مقاله صفات QoS برای هر یک از سرویس‌های $ws_{i,j}$ به صورت مقابل است:

$$q(ws_{i,j}) = \{q_{rt}, q_a, q_t, q_p, q_{re}\}$$

معنای خاص هر کدام از صفات به صورت زیر تعریف می‌شود:

زمان پاسخ‌دهی q_{rt} (*Response Time*): زمان پاسخ‌دهی بازه ای است بین ارسال درخواست و دریافت پاسخ از $w_{si,j}$ است.

موجودیت q_a (*Availability*): در واقع احتمال فراخوانی موفق را نشان می‌دهد.

توان عملیاتی q_t (*Throughput*): میزان کل فراخوانی سرویس $w_{si,j}$ در یک بازه زمانی را می‌گویند.

هزینه q_p (*Price*): هزینه‌ای که به فراهم کنند سرویس پرداخت می‌شود توسط درخواست‌کننده سرویس اندازه‌گیری می‌شود و به واحد پول مثلا دلار است.

تکرار q_{re} (*Reputation*): تکرار در سرویس یکی از ویژگی‌های مهم تشخیص سرویس مورد اعتماد است. مقدار تکرار برخلاف صفات دیگر از طریق پاسخ کاربران بدست می‌آید بنابراین ممکن است در شرایط مختلف متفاوت باشد. مثلا زمانی که سرویس‌ها بهبود داده می‌شوند کاربران بیشتر مورد تاثیر قرار می‌گیرند و بیشتر جذب می‌شوند.

جدول ۳: توابع تجمیع در ترکیب سرویس ابری (QoS)

عملیات	نوع صفت کیفیت	تابع تجمیع
جمع	زمان پاسخ‌دهی q_{rt}	$Q_{rt} = \sum_{i=1}^m q_{rt}(w_{si,j})$
ضرب	موجودیت q_a	$Q_a = \prod_{i=1}^m q_a(w_{si,j})$
مینیمم	توان عملیاتی q_t	$Q_t = \min_{i=1}^m q_t(w_{si,j})$
جمع	هزینه q_p	$Q_p = \sum_{i=1}^m q_p(w_{si,j})$
میانگین	تکرار q_{re}	$Q_{re} = \frac{1}{m} \sum_{i=1}^m q_{re}(w_{si,j})$

۳-۱- پیاده سازی مسئله ترکیب سرویس ابری

در ترکیب‌سازی سرویس‌ها، نوع صفات چون در هر سرویس انتزاعی با سرویس انتزاعی دیگر یکسان هستند هر نوع صفت برای خود یک تابع تجمیع (*Aggregation*) دارد که برای تجمیع کردن آن صفت برای همه سرویس‌های انتزاعی استفاده می‌کنیم توابع تجمیع برای هر صفت در جدول (۳) مشخص شده است. هر صفت دارای یک درجه اهمیت است که با یک متغیر وزنی مشخص می‌شود. مجموع وزن صفات باید برابر مقدار یک باشد یعنی $\sum_{k=1}^r w_k = 1$ که در اینجا w_k وزن صفت k ام می‌باشد. وزن‌ها می‌تواند توسط کاربر مشخص شود و یا براساس اهمیت تنظیم شود. از آنجا که صفات قرار است به یک وزن مشخص ضرب شوند بهتر است قبل از ضرب مقادیر کل مقادیر صفات نرمال‌سازی شود. هر صفت می‌تواند مفهوم مثبت یا منفی داشته باشد به معنای این که هر چقدر صفت مثبت بیشتر باشد باعث افزایش کیفیت و هر چقدر صفت منفی بیشتر باشد باعث

کاهش کیفیت سرویس می‌شود. برای مثال، قیمت (*Price*)، یک صفت منفی است، یعنی باید مقدار آن کم باشد تا کیفیت سرویس بهتر شود. صفات مثبت و منفی به صورت جدا نرمال‌سازی می‌شوند که توسط دو فرمول (۱) و (۲) مشخص شده است.

نرمال‌سازی صفات مثبت:

$$UniQ_k = \begin{cases} \frac{Q_k - \min Q_k}{\max Q_k - \min Q_k} & , if \max Q_k \neq \min Q_k \\ 1 & , if \max Q_k = \min Q_k \end{cases} \quad (1)$$

نرمال‌سازی صفات منفی:

$$UniQ_k = \begin{cases} \frac{\max Q_k - Q_k}{\max Q_k - \min Q_k} & , if \max Q_k \neq \min Q_k \\ 1 & , if \max Q_k = \min Q_k \end{cases} \quad (2)$$

که در اینجا $\max Q_k$ و $\min Q_k$ به ترتیب نشان‌دهنده بیشترین و کمترین مقدار برای صفت در بعد k ام برای همه حالات ترکیب است.

تابع هدف مسئله ترکیب سرویس ابری می‌تواند به صورت معادله (۳) بیان شود:

$$\max csQoS = \sum_{k=1}^r UniQ_k * w_k \quad (3)$$

s. t

$$1 \leq i \leq m, i \in Z$$

$$1 \leq j \leq n_i, j \in Z$$

$$1 \leq k \leq r, k \in Z$$

۴- روش پیشنهادی

۴-۱- الگوریتم کرم شب تاب نوین

الگوریتم تغییر یافته همانطور که مطرح شد برگرفته از آخرین بررسی‌هایی می‌باشد که در مورد رفتارهای کرم شب‌تاب صورت گرفته است. نمودار جریان شکل (۲) مراحل الگوریتم را نشان می‌دهد. در بخش‌های بعد به توضیح مرحله به مرحله الگوریتم می‌پردازیم.

۴-۱-۱- مرحله اول الگوریتم

در الگوریتم کرم شب‌تاب جدید مانند الگوریتم قبلی ابتدا با مقداردهی متغیرهای الگوریتم، تعیین جمعیت اولیه و شرط پایان تکرار الگوریتم شروع می‌شود ابتدا متغیرهای الگوریتم کرم شب‌تاب (γ ضریب جذب نور، β ضریب جذابیت پایه و α ضریب تصادفی) بعلاوه دو متغیر جدید $P_{parasitism}$ و θ که به ترتیب میزان احتمال تغذیه شدن کرم شب‌تاب و شدت تمایل کرم شب‌تاب برای حرکت به سمت جمع را مشخص می‌کند مقداردهی اولیه می‌شود. در حین تعیین جمعیت اولیه بهترین جواب را نیز از میان جمعیت اولیه مشخص می‌کنیم.

۴-۱-۲- مرحله دوم الگوریتم

تکرار اصلی الگوریتم شروع شده و مانند الگوریتم کرم شب‌تاب قبلی ولی با این تفاوت که با متغیر تصادفی $P_{parasitism}$ احتمال شکارچی بودن و نبودن کرم شب‌تاب را سنجش می‌کنیم برا کنترل این امر از تولید یک عدد تصادفی استفاده کرده و با یک شرط *if* بررسی می‌کنیم که آیا این مقدار از مقدار احتمال $P_{parasitism}$ بزرگتر است یا خیر در صورتی که بزرگتر باشد دیگر شکارچی نبوده وارد حالت عادی که تغییراتی در تابع بروز رسانی آن هم صورت گرفته است می‌شویم ولی در صورتی که کوچکتر باشد وارد فاز شکارچی شده و عملیات بروز رسانی مرتبط با آن صورت می‌گیرد.

۴-۱-۲-۱- فاز شکارچی

در رفتار شکارچی کرم شب‌تاب همان‌طور که قبلاً توضیح داده شده است، تابع بروز رسانی به صورت رابطه (۴) پیاده سازی می‌شود.

$$X_{new} = Mutate(X) \quad (4)$$

در این فاز چون قرار است کرم شب‌تاب به طور کلی حذف شود از کرم شب‌تاب قربانی با استفاده از تابع جهشی که به صورت فرمول (۶) تعریف می‌شود اقدام به ساخت راه حل جدید می‌کنیم در صورتی که راه حل جدید از کرم شب‌تاب قربانی بهتر باشد جایگزین آن راه حل می‌شود و همچنین جواب تولید شده در صورت بهتر بودن از بهترین جواب الگوریتم $BestSol$ نیز با مقدار جهش یافته جایگزین می‌شود.

۴-۱-۲-۲- تابع جهش

تابع جهش استفاده شده در فاز شکارچی به صورت فرمول (۶) تعریف می‌شود.

$$\sigma = \frac{Var_{max} - Var_{min}}{10} \quad (5)$$

$$\begin{aligned} X_{paratism} &= X, \\ X_{paratism}(j) &= X(j) + \sigma \cdot rand \end{aligned} \quad (6)$$

بر اساس معادله (۶) تابع جهش بدین صورت به کرم شب‌تاب قربانی اعمال می‌شود.

$$X_{paratism} = Mutate(X_i) \quad (7)$$

۴-۱-۲-۳- فاز بروزسانی تولید مثل تغییر یافته بر پایه بهترین جواب

در صورتی که کرم شب‌تاب شکارچی نباشد وارد این فاز می‌شویم. در این فاز روش همانند حالت عادی بروزسانی می‌باشد فقط یک بخش جدید به تابع بروزسانی اضافه می‌شود تا تمایل حرکت کرم شب‌تاب به سمت جمع را کنترل کنیم. تابع بروزسانی به صورت فرمول (۱۱) تغییر می‌یابد.

$$\beta = \beta_0 * e^{-\gamma * r_{ij}^m} \quad (8)$$

$$\epsilon = \delta * rand(-1,1) \quad (9)$$

$$\delta = 0,05 * (Var_{Max} - Var_{Min}) \quad (10)$$

$$X_{new} = X_i + \beta * rand * (X_j - X_i) + \theta * rand * (BestSol - X_i) + \alpha * \epsilon \quad (11)$$

در این فرمول پارامترهای β ، α و γ همان پارامترهای الگوریتم کرم شب‌تاب می‌باشد r_{ij} فاصله اقلیدسی [28] دو کرم شب‌تاب، $BestSol$ بهترین جواب است. ϵ و δ عبارت‌هایی است که نحوه محاسبه آنها در فرمول‌های (۹) و (۱۰) بیان شده است. برای کنترل حرکت به سمت جمع پارامتر θ جهت کنترل میزان حرکت به سمت جمع در ابتدای الگوریتم مقاردهی می‌شود. با این تغییر، دیگر کرم شب‌تاب فقط تحت تاثیر کرم شب‌تاب پر نور نبوده و علاوه بر حرکت به سمت کرم شب‌تاب پر

نور به سمت جمع نیز حرکت می‌کند. بعد از تولید جواب جدید توسط تابع بروزرسانی تغییر یافته، جواب جدید را با هزینه جمعیت جدید کرم شب تاب I مقایسه و در صورتی که از آن بهتر باشد جایگزین آن می‌شود و همچنین در صورت بهتر بودن از $BestSol$ نیز جایگزین آن می‌شود.

جمعیت جدید کرم شب تاب با جمعیت قبلی الحاق شده بهترین جمعیت را به اندازه جمعیتی که مشخص است یعنی ($nPop$) جدا کرده و برای تکرار بعدی الگوریتم به کار می‌بریم. شرط پایان را چک می‌کنیم در صورتی که برقرار باشد تکرار پایان می‌پذیرد در غیر اینصورت تکرار دیگری را با بهترین جمعیت جدا شده، آغاز می‌کنیم.
الگوریتم کرم شب تاب تغییر یافته به صورت زیر می‌باشد:

شبه کد روش مطرح شده (NFA):

- 1: **Input:** objective function f , constraints and the dimensions of the problem (D)
- 2: Initialization
- 3: Initial parameters $\theta, nPop, P_{paratism}, \alpha, \gamma, \beta$
- 4: Let pop be the set of firefly $X_1, X_2, \dots, X_{nPop}$

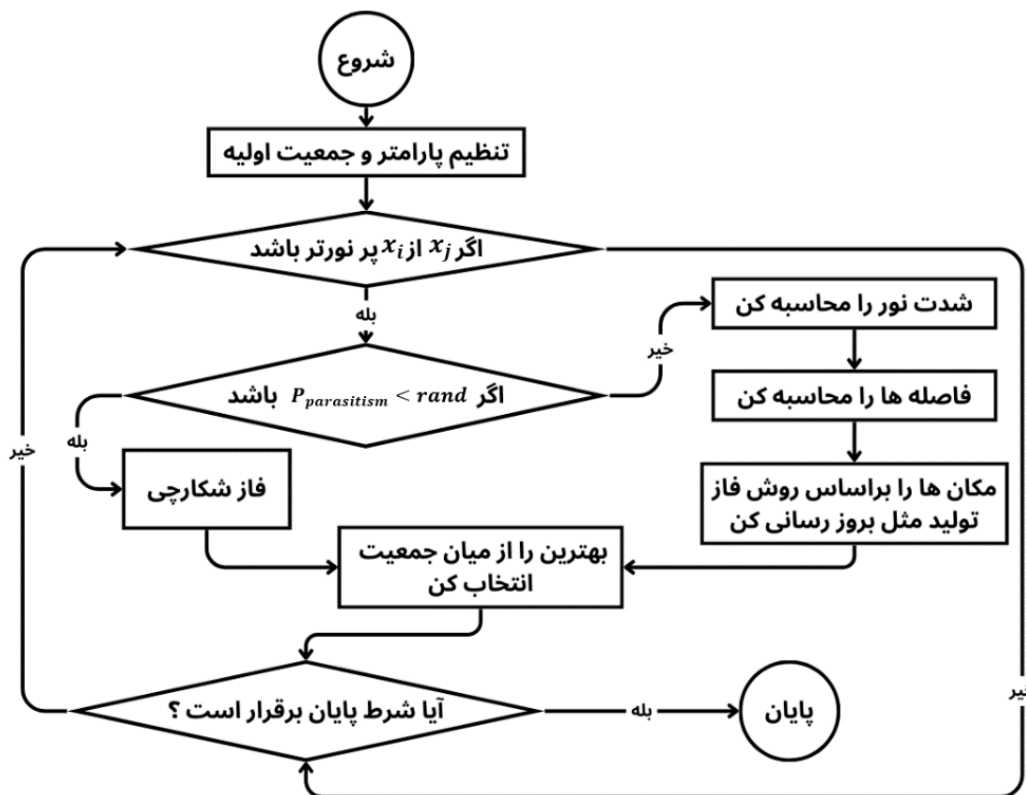
$$pop = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{nPop} \end{bmatrix} \quad X_i \text{ is firefly sample}$$
- 5: **for** each of firefly sample in pop **do**
- 6: Assign random real number between [$VarMin, VarMax$] to the firefly sample position
- 7: Calculate Cost for firefly sample position

$$X = [X.cost \quad X.position]$$
- 8: **end for**
- 9: Get best firefly from pop as $BestSol$
- 10: Iterations
- 11: **while** (the stopping criterion is not met) **do**
- 12: **for** each of firefly i **do**
- 13: **for** each of firefly j **do**
- 14: **if** $X_j < X_i$
- 15: calculate distance and light intensity.
- 16: **if** $P_{paratism} < rand$
- 17: calculate $X_{paratism}$ for X_i with Eq. (7)
- 18: **if** $X_{paratism} < X_i$
- 19: $X_i = X_{paratism}$
- 20: **if** $X_i < BestSol$
- 21: $BestSol = X_i$
- 22: **end if**
- 23: **end if**
- 24: **else**
- 25: calculate X_{new} from X_i by Eq. (11)
- 26: **if** $X_{new} < newpop_i$
- 27: $newpop_i = X_{new}$
- 28: **if** $newpop_i < BestSol$

```

29:         BestSol = newpopi
30:     end if
31: end if
32: end if
33: end if
34: end for
35: end for
36: Concatenate pop with newpop and select best population with size of nPop
37: decrease the  $\alpha$  ( $\alpha * \alpha\_damp$ )
38: end while
39: \\the final stage
40: output the optimum founded value

```



شکل ۲: نمودار جریان الگوریتم مطرح شده NFA

۵- ارزیابی و نتایج

در این بخش، روش بهبود یافته الگوریتم کرم شب‌تاب را مورد ارزیابی قرار می‌دهیم و در کل نتایج بدست آمده هر جدول

از کامپیوتری با مشخصات حافظه اصلی ۸ گیگابایت، سیستم عامل ویندوز ۶۴ بیتی و پردازنده *Core i7* اینتل استفاده شده است و ارزیابی روش مطرح شده به سه بخش تقسیم می‌شود:

توابع استاندارد بهینه سازی: برای ارزیابی این الگوریتم ۲۶ عدد تابع بهینه سازی استاندارد از نوع *unimodal* و *multimodal* در بُعد های ۲، ۱۰، و ۳۰ تعریف شده است تا عملکرد الگوریتم مطرح شده از لحاظ همگرایی بررسی شود. **مجموعه داده‌های QWS:** این مجموعه داده توسط *Eyhab al-masri* [29] گردآوری شده است که شامل ۲۵۰۷ داده واقعی سرویس‌های وب است که هر کدام دارای ۹ پارامتر کیفیت به ترتیب *Throughput*، *Successability*، *Reliability*، *Responsetime*، *Availability* و ... می‌باشد. اخیراً این داده‌ها در بسیاری از مقالات مانند [30] و [31] مورد استفاده قرار گرفته است.

داده‌های تصادفی: از آنجا که تعداد و تنوع داده‌ها کم است در بسیاری از تحقیق‌ها از داده‌های تصادفی برای بررسی عملکرد الگوریتم نیز استفاده می‌شود. برای اینکه سرویس‌های ابری بزرگ مقیاس را شبیه‌سازی کنیم به تعداد ۵۰۰،۰۰۰ سرویس به صورت تصادفی تولید شده است که دارای ۴ پارامتر کیفیت *Price*، *Responsetime*، *Availability* و *Reputation* می‌باشد. مقادیر این داده‌های تصادفی در بین بازه بسته $[0, 1]$ به صورت توزیع یکنواخت و تصادفی تولید شده اند.

برای ارزیابی این الگوریتم ابتدا ۲۶ عدد تابع بهینه سازی استاندارد مورد ارزیابی قرار می‌گیرد، روش مطرح شده با الگوریتم‌های معروف *FA*، *PSO*، *ABC* و الگوریتم‌های دیگر جدید (۲۰۲۲ تا ۲۰۲۴) فرا ابتکاری *WO*، *ACVO*، *FFO*، *EVO*، *CDO* و *SO* مقایسه می‌شود. آزمون‌های بدون پارامتر *Friedman* و *Pairwise* [32] اجرا می‌شود و نمودارهای ارزیابی سرعت همگرایی *CPU-Time*، برای نتایج بدست آمده ارزیابی و رسم می‌شود. برتری الگوریتم نسبت به روش‌های پیشین با بررسی این نتایج مشخص می‌شود. توابع ارزیابی استاندارد دارای دو حالت تک مُد (*unimodal*) و چند مُد (*multimodal*) در بُعد‌های مختلف هستند که در دو جدول (۴) و (۵) به ترتیب توصیف شده اند. منظور از مُد همان تعداد قله در توزیع تابع می‌باشد. برای تنظیم پارامترها از جدول (۶) استفاده کرده‌ایم و نتایج آماری بدست آمده حاصل ۳۰ بار اجرای هر تابع توسط الگوریتم‌ها می‌باشد. شرط پایان برای اجرا مدت ۱۰ ثانیه اجرا ($CPU - Time = 10$) می‌باشد.

با بررسی جدول‌های (۷) و (۸) مشخص می‌شود که روش مطرح شده همگرایی دقیقی داشته و در عین داشتن دقت بالا توانایی رسیدن به جواب‌های بهتر را نیز از خود نشان می‌دهد. برای مثال در توابع *F13*، *F9*، *F4*، *F22*، *F21*، *F19*، *F16*، *F15* و *F1* توانسته به مقدار کمینه 0 برسد یعنی بتواند تا دقت *e-330* آن توابع را کمینه سازی کند در این جدول *Best* (بهترین)، *Worst* (بدترین جواب)، *Mean* (میانگین) و *Std* (انحراف معیار) نتایج آماری بدست آمده از ۳۰ بار اجرا برای تابع مورد نظر می‌باشد. روش مطرح شده در بهینه سازی نوع توابع چند مُد (*multimodal*) خیلی بهتر نسبت به بقیه روش‌ها عمل می‌کند و این نشان می‌دهد که الگوریتم *NFA* می‌تواند در حالت‌های که توزیع تابع به صورت چند قله و ناهموار است عملکرد بهتری را نشان دهد.

با بررسی نمودارهای همگرایی شکل (۳) برای توابع $F1, F4, F9, F13, F18, F19, F20, F25, F26$ مشخص است که روش مطرح شده دارای پیچیدگی زمانی پایینی بوده و توانسته است در اکثر توابع بخصوص در تابع $F4, F9, F13$ ، $F20$ و $F26$ با سرعت بسیار بالایی همگرا به بهترین جواب شود که روش های دیگر قادر نبودند. در این نمودارها محور افقی زمان ($CPU-Time$) می باشد و محور عمودی نشان دهنده بهترین جوابی است که در آن زمان بدست آمده است. با اینکه شرط پایان اجرا ۱۰ ثانیه تنظیم شده است ولی روش مطرح شده در اکثر توابع در مدت کمتر از ۲ ثانیه به مقدار بهترین جواب رسیده است. با بررسی شیب همگرایی نمودار NFA با شیب همگرایی نمودار FA مشخص است که الگوریتم کرم شبتاب مطرح شده از نظر همگرایی خیلی سریعتر نسبت به قبل شده است. الگوریتم های $ACVO$ و FFO نیز سرعت همگرایی خوبی را از خود نشان می دهند. الگوریتم های SO و EVO از لحاظ سرعت همگرایی خوب عمل نکرده اند.

جدول ۴: مشخصات توابع استاندارد بهینه سازی unimodal

نوع		تابع	D	دامنه	کمینه	فرمول
	F1	Beale	2	[-4.5,4.5]	0	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$
	F2	Easom	2	[-100,100]	-1	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
	F3	Matyas	2	[-10,10]	0	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
	F4	Colville	4	[-10,10]	0	$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$
UN	F5	Zakharov	10	[-5,10]	0	$f(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5ix_i)^2 + (\sum_{i=1}^D 0.5ix_i)^4$
	F6	Schwefel 2.22	30	[-10,10]	0	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $
	F7	Schwefel 1.2	30	[-100,100]	0	$f(x) = \sum_{i=1}^D (\sum_{j=1}^i X_j)^2$
	F8	Dixon-Price	30	[-10,10]	0	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$
	F9	Step	30	[-5.12, 5.12]	0	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$
US	F10	Sphere	30	[-100,100]	0	$f(x) = \sum_{i=1}^D x_i^2$
	F11	SumSquares	30	[-10,10]	0	$f(x) = \sum_{i=1}^D ix_i^2$
	F12	Quartic	30	[-1.28, 1.28]	0	$f(x) = \sum_{i=1}^D ix_i^4 + Rand$

برای اینکه بتوانیم به صورت غیر وابسته به نوع توابع و الگوریتم‌ها روش‌های مطرح شده امتیاز دهی کنیم از روش آزمون *Friedman* برای امتیاز دهی الگوریتم‌های مطرح شده استفاده کردیم. آزمون *Friedman* یک آزمون بدون پارامتر، معادل ارزیابی واریانس با اندازه‌های تکراری (درون گروهی است) که از آن برای مقایسه میانگین رتبه‌ها در بین k متغیر (گروه) استفاده می‌شود. همان طور که از شکل (۴) مشخص است در این آزمون روش مطرح شده توانسته است بهترین امتیاز یعنی ۳,۸۳ را اذعان خود کند.

برای مقایسه دو به دو الگوریتم‌ها از آزمون دوگانه (*Pairwise*) استفاده شده است. نتایج مربوط به این آزمون در جدول (۹) نمایش داده شده است. در این جدول مقدار *Sig* هر چقدر کمتر باشد نشان از تفاوت بسیار آن دو الگوریتم می‌باشد. در این آزمون مقدار آستانه ۰,۰۵ تنظیم شده است. با بررسی نتایج مشخص است که نتایج روش مطرح شده به الگوریتم *ACVO* نزدیک و از الگوریتم *EVO* بسیار متفاوت (بهتر) می‌باشد. از جدول مشخص است که الگوریتم مطرح شده از روش‌های *FFO*, *EVO* و *CDO* بسیار بهتر عمل می‌کند چون مقدار *Sig* در این الگوریتم‌ها از مقدار آستانه ۰,۰۵ پایین می‌باشد.

جدول ۵: مشخصات توابع استاندارد بهینه سازی multimodal

نوع	تابع	D	دامنه	کمینه	فرمول
	F13 Schaffer	2	[-100,100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$
	F14 6 H Camel	2	[-5,5]	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
	F15 Boachevsky2	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1)(4\pi x_2) + 0.3$
	F16 Boachevsky3	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$
MN	F17 Shubert	2	[-10,10]	-186.73	$f(x) = (\sum_{i=1}^5 i \cos((i+1)x_1 + i)) (\sum_{i=1}^5 i \cos((i+1)x_2 + i))$
	F18 Rosenbrock	30	[-30,30]	0	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
	F19 Griewank	30	[-600,600]	0	$f(x) = \frac{1}{4000} (\sum_{i=1}^D (x_i - 100)^2) - (\prod_{i=1}^D \cos(\frac{x_i - 100}{\sqrt{i}})) + 1$
	F20 Ackley	30	[-32,32]	0	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$
	F21 Bohachevsky 1	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$
	F22 Booth	2	[-10,10]	0	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
	F23 Michalewicz 2	2	[0,π]	-1.8013	$f(x) = -\sum_{i=1}^D \sin(x_i) (\sin(\frac{ix_1^2}{\pi}))^{20}$

MS	F24	Michalewicz 5	5	$[0, \pi]$	-4.6877	$f(x) = - \sum_{i=1}^D \sin(x_i) (\sin(\frac{ix_i^2}{\pi}))^{20}$
	F25	Michalewicz 10	10	$[0, \pi]$	-9.6602	$f(x) = - \sum_{i=1}^D \sin(x_i) (\sin(\frac{ix_i^2}{\pi}))^{20}$
	F26	Rastrigin	30	$[-5.12, 5.12]$	0	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$

جدول ۶: تنظیمات پارامترهای الگوریتم های مطرح شده

الگوریتم	منبع	سال	تنظیمات پارامتر
FA	[3]	2009	$N = 30, \alpha = .2, \beta_0 = 2, \gamma = 2$
PSO	[7]	2010	$N = 30, C_1, C_2 = 2, W = 1$
ABC	[6]	2010	$N = 50, Non - looker = 50, \alpha = 0.1, L = N * D * .6$
GWO	[8]	2014	$N = 30, a(2 \rightarrow 0)$
WOA	[9]	2016	$N = 30, a(2 \rightarrow 0), a(-1 \rightarrow -2)$
ACVO	[14]	2022	$N = 25, R_0 = 2.5, \Delta = 2, qd = 5, hd = 10, r1, r2 \in [0, 0.5]$
SO	[13]	2022	$N = 30$
FFO	[17]	2023	$a = 1.2, pa = 0.05$
EVO	[16]	2023	$N = 20$
CDO	[15]	2023	$N = 30, S_\gamma = rand(1,300,000), S_\beta = rand(1,270,000), S_\alpha = rand(1,16,000), r = rand(0,1)$
WO	[12]	2024	$N = 100, P = 0.45, T = 2000$
NFA	-	-	$N = 30, \alpha = .2, \beta_0 = 2, \gamma = 2, \theta = 1.5, P_{paratism} = 0.9$

جدول ۷: مقایسه نتایج بدست آمده الگوریتم ها در حل توابع استاندارد بهینه سازی unimodal

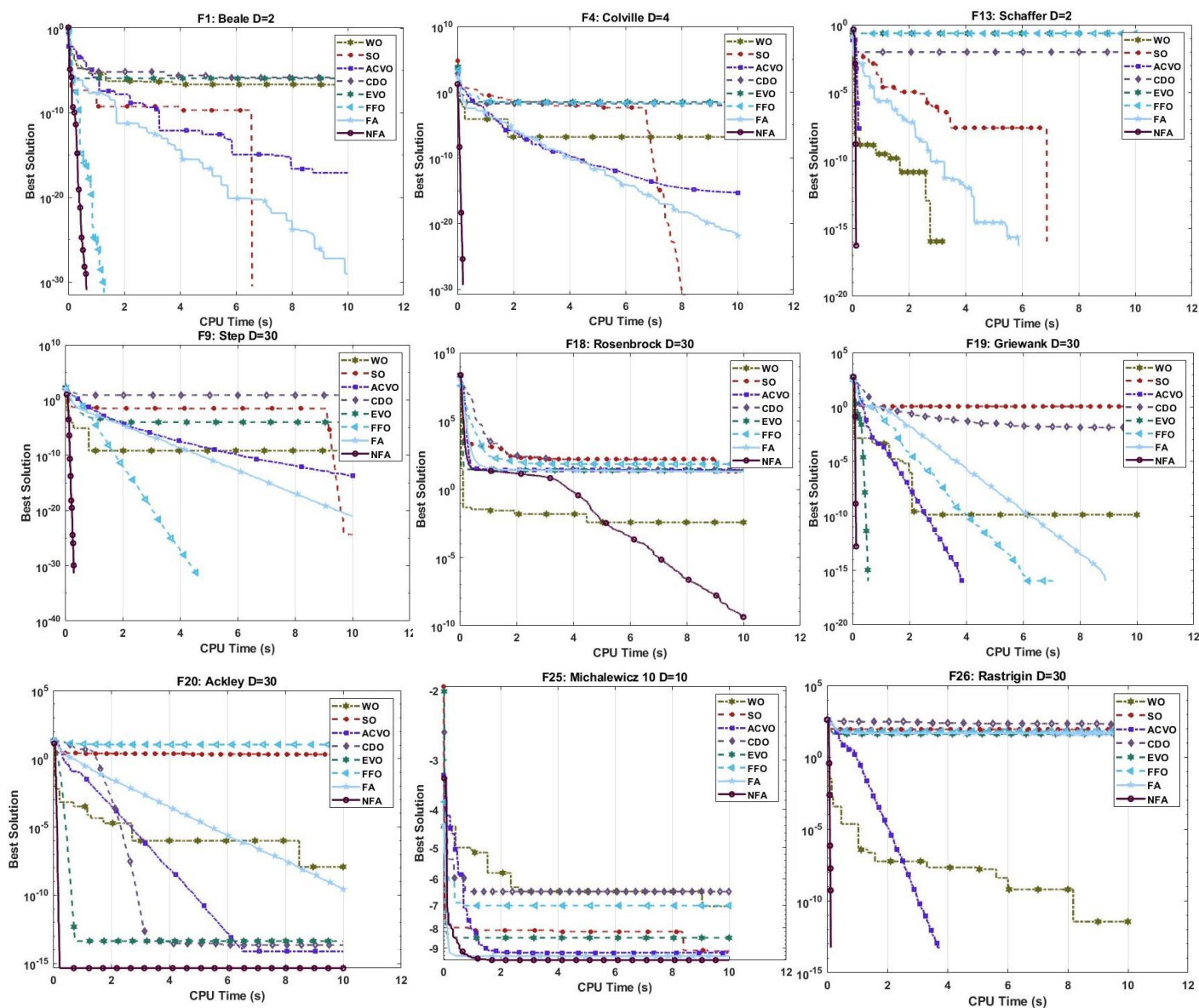
تابع	PSO	FA	ABC	FFO	EVO	CDO	ACVO	SO	WO	NFA
F1 / Mean	5.045437	2.74E-28	9.90E-15	0	0.4747572	9.14E-07	3.05E-18	0	1.07E-07	0
Best	5.045437	3.75E-30	1.20E-17	0	4.01E-06	2.86E-08	9.63E-20	0	6.69E-09	0
Worst	5.045437	9.03E-28	7.52E-14	0	2.0265911	2.12E-06	1.20E-17	0	3.75E-07	0
Std	9.36E-16	3.06E-28	2.31E-14	0	0.7672896	8.18E-07	3.70E-18	0	1.39E-07	0
F2 / Mean	-1	-1	-1	-0.90001	-1.99E-01	-0.99999	-1	-1	-1	-1
Best	-1	-1	-1	-1	-1.00E+00	-1	-1	-1	-1	-1
Worst	-1	-1	-1	-8.11E-05	0	-0.99998	-1	-1	-1	-1
Std	0	0	0	0.316202	4.20E-01	5.00E-06	5.23E-17	0	3.47E-07	0
F3 / Mean	0	3.14E-29	2.94E-15	0	0.0109007	0	0	0	5.93E-20	3.27E-42
Best	0	2.79E-30	2.34E-19	0	9.59E-14	0	0	0	1.11E-25	1.47E-46
Worst	0	1.18E-28	2.88E-14	0	0.0606005	0	0	0	5.56E-19	3.27E-41
Std	0	3.35E-29	9.09E-15	0	0.0220548	0	0	0	1.75E-19	1.03E-41
F4 / Mean	9.16E-07	7.21E-24	0.160557	0.002628	3.2215448	0.012096	2.94E-12	3.73E-31	1.11E-06	1.77E-32
Best	1.16E-07	2.11E-24	0.03255	1.13E-06	0.072911	0.00301	3.22E-22	0	1.83E-09	0
Worst	2.11E-06	2.25E-23	0.311078	0.017076	7.7431727	0.022109	2.94E-11	2.84E-30	7.62E-06	1.77E-31
Std	6.76E-07	5.62E-24	0.093514	0.005173	2.3927719	0.006269	9.30E-12	8.95E-31	2.33E-06	5.61E-32
F5 / Mean	1.78E-122	6.20E-24	0.864602	9.17E-39	17.20844	4.9E-324	3.27E-30	4.874e-317	1.93E-14	2.93E-38
Best	4.06E-131	3.18E-24	0.236182	4.26E-43	1.74E-25	4.900e-324	4.63E-31	0	2.93E-19	7.07E-41
Worst	1.74E-121	9.61E-24	1.804045	3.27E-38	135.84496	4.900e-324	1.20E-29	4.873e-316	1.21E-13	2.9E-37
Std	5.49E-122	2.31E-24	0.53746	9.47E-39	42.340943	0	4.09E-30	0	3.88E-14	9.17E-38
F6 / Mean	0.000116	5.69E-11	4.39E-11	2.14E-42	0.040037	0	9.82E-24	8.05E-11	1.81E-06	4.25E-19
Best	3.13E-07	4.78E-11	1.46E-11	4.99E-43	1.11E-35	0	6.44E-24	8.23E-17	8.23E-09	3.15E-19
Worst	0.000818	7.18E-11	7.53E-11	4.35E-42	0.3995744	0	1.25E-23	7.40E-10	1.11E-05	7E-19
Std	0.000252	6.75E-12	1.72E-11	1.17E-42	0.1263288	0	2.04E-24	2.32E-10	3.44E-06	1.14E-19
F7 / Mean	992.941	2548.319	71045.42	16849.38	82564.486	3143159	0.164587	8448.495	4.15E-07	231.0862
Best	230.6411	1112.782	55534.36	11068.04	3776.3192	1.74E-100	0.010988	3542.206	8.20E-12	38.56414
Worst	1610.294	3459.261	94933.48	26292.35	218416.53	8696810	0.590002	14677.38	2.99E-06	610.7664
Std	386.9328	609.3132	13132.97	5149.582	74811.696	3111697	0.209673	4066.877	9.36E-07	165.157
F8 / Mean	0.672121	0.666667	0.004559	1.156206	0.6669802	0.666667	0.666667	0.666667	0.24908	0.666667
Best	0.666667	0.666667	0.003475	0.666667	0.666667	0.666667	0.666667	0.666667	0.245672	0.666667
Worst	0.721195	0.666667	0.007742	3.510627	0.6689086	0.666667	0.666667	0.666667	0.249477	0.666671
Std	0.017243	0	0.001654	0.99817	0.0007106	0	0	0	0.001198	1.17E-16
F9 / Mean	1.49E-31	4.18E-23	4.13E-22	0	0.0003216	7.5	1.75E-14	3.99E-20	2.73E-08	2.05E-33
Best	0	2.35E-23	2.96E-23	0	4.74E-05	7.5	1.14E-14	9.90E-25	5.30E-11	0
Worst	6.13E-31	6.49E-23	2.40E-21	0	0.0014192	7.5	2.29E-14	3.70E-19	8.92E-08	2.47E-32
Std	2.11E-31	1.27E-23	7.28E-22	0	0.0004011	5.92E-16	3.62E-15	1.16E-19	3.29E-08	5.45E-33
F10 / Mean	9.60E-125	1.37E-20	3.09E-16	1.11E-76	1.34E-08	0	1.24E-46	2.94E-11	1.35E-10	6.2E-38
Best	6.69E-138	6.64E-21	3.24E-17	9.26E-78	1.60E-260	0	3.76E-48	7.05E-35	5.85E-19	7.81E-40
Worst	8.87E-124	1.91E-20	1.89E-15	4.35E-76	1.34E-07	0	8.46E-46	2.94E-10	7.80E-10	7.23E-38
Std	2.79E-124	3.99E-21	5.63E-16	1.36E-76	4.25E-08	0	2.55E-46	9.30E-11	2.58E-10	1.55E-38
F11 / Mean	1.22E-122	2.05E-21	1.96E-18	8.03E-77	0.14178	0	8.14E-46	2.15E-16	5.03E-12	4.03E-39
Best	4.09E-140	1.52E-21	6.54E-20	9.73E-79	0	0	2.44E-46	4.91E-36	5.36E-17	1.04E-40
Worst	1.01E-121	3.44E-21	4.84E-18	2.26E-76	1.4178	0	2.27E-45	2.15E-15	4.57E-11	1.41E-38
Std	3.20E-122	6.12E-22	2.03E-18	8.00E-77	0.4483477	0	5.71E-46	6.80E-16	1.43E-11	3.21E-39
F12 / Mean	0.005222	0.002214	0.301523	0.010861	0.029599	0.000164	0.003686	0.001217	8.68E-05	0.003892
Best	0.002837	0.000994	0.250508	0.003183	0.0073246	7.87E-05	4.96E-05	0.000934	2.99E-05	0.00234
Worst	0.007685	0.004082	0.377448	0.020635	0.0459311	0.000294	0.010265	0.001607	0.000176	0.006956
Std	0.001595	0.000913	0.040942	0.005228	0.013058	7.75E-05	0.003134	0.000216	4.49E-05	0.001897

جدول ۸: مقایسه نتایج بدست آمده الگوریتم ها در حل توابع استاندارد بهینه سازی multimodal

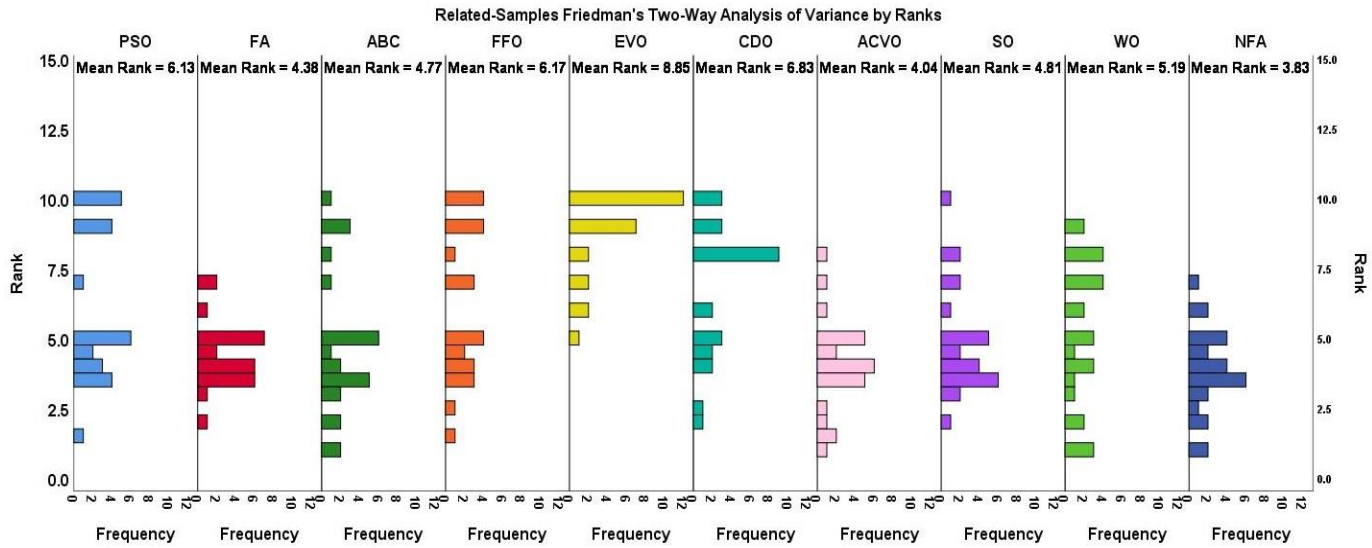
تابع	PSO	FA	ABC	FFO	EVO	CDO	ACVO	SO	WO	NFA
F13 / Mean	0	4.79E-08	0.063464	0.0883607	0.008793	0	0	3.33E-17	0	0
Best	0	0	0.009716	0.0097159	0.00049	0	0	0	0	0
Worst	0	3.67E-07	0.22769	0.2652649	0.009716	0	0	2.78E-16	0	0
Std	0	1.17E-07	0.077444	0.0847698	0.002917	0	0	8.76E-17	0	0
F14 / Mean	-1.03163	-1.03163	-1.03163	-0.950091	-1.00316	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
Best	-1.03163	-1.03163	-1.03163	-1.031628	-1.0316	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
Worst	-1.03163	-1.03163	-1.03163	-0.388121	-1	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
Std	0	0	0	0.2008668	0.009993	0	0	1.04E-07	0	2.34E-16
F15 / Mean	0	0	0	2.7023915	0	0	0	6.11E-17	0	0
Best	0	0	0	3.23E-06	0	0	0	0	0	0
Worst	0	0	0	20.718434	0	0	0	5.00E-16	0	0
Std	0	0	0	6.5180783	0	0	0	1.56E-16	0	0
F16 / Mean	0	5.42E-11	0	24.280313	0	0	0	4.60E-15	0	0
Best	0	5.55E-17	0	3.73E-06	0	0	0	0	0	0
Worst	0	5.42E-10	0	228.39491	0	0	0	4.31E-14	0	0
Std	0	1.71E-10	0	71.848864	0	0	0	1.36E-14	0	0
F17 / Mean	-186.73	-186.73	-159.195	-171.8515	-180.396	-186.73	-186.73	-186.728	-186.73	-186.731
Best	-186.73	-186.73	-186.731	-186.7308	-186.729	-186.73	-186.73	-186.731	-186.73	-186.731
Worst	-186.73	-186.73	-37.6811	-107.6902	-123.53	-186.73	-186.73	-186.713	-186.73	-186.731
Std	0	0	50.12255	25.88913	19.98075	0	0	0.005696	0	1.34E-14
F18 / Mean	32.40313	0.378283	62.40886	26.267995	25.56692	27.02315	19.7458	0.000135	25.17723	9.84E-07
Best	10.33688	0.081335	20.50847	24.756371	25.34383	26.97153	16.08809	5.29E-06	19.64031	6.58E-08
Worst	79.1351	1.105278	86.33429	28.766824	26.09291	27.15731	24.72807	0.000719	70.99525	4.4E-06
Std	23.79103	0.289675	28.37315	1.4277663	0.226316	0.066769	2.279785	0.000211	16.10541	1.29E-06
F19 / Mean	0.022837	1.91E-14	0	0.0015035	0.008071	0	0.48655	8.61E-10	0.002709	0.001725
Best	0	2.22E-16	0	0	0	0	0.002712	2.31E-13	0	0
Worst	0.053831	6.41E-14	0	0.0150346	0.010396	0	1.138093	5.07E-09	0.01969	0.009857
Std	0.017193	2.43E-14	0	0.0047544	0.003071	0	0.548548	1.66E-09	0.006403	0.003683
F20 / Mean	1.54E-14	1.95E-09	12.40333	0.0017596	2.036083	1.18E-14	4.15E-05	1.63E-06	9.19E-11	2.22E-14
Best	1.47E-14	9.50E-10	8.085596	1.47E-14	1.47E-14	7.55E-15	7.84E-09	1.80E-08	7.70E-11	2.22E-14
Worst	2.18E-14	3.80E-09	16.76429	0.0175819	20.36083	1.47E-14	0.000411	6.59E-06	1.11E-10	2.22E-14
Std	2.25E-15	7.80E-10	3.173442	0.0055594	6.438658	3.26E-15	0.00013	2.03E-06	1.10E-11	0
F21 / Mean	0	0	0	3.731573	0	0	0	1.18E-15	0	0
Best	0	0	0	4.22E-11	0	0	0	0	0	0
Worst	0	0	0	32.119791	0	0	0	1.13E-14	0	0
Std	0	0	0	10.074071	0	0	0	3.57E-15	0	0
F22 / Mean	1.8	0	0	1.2753229	4.52E-06	1.75E-16	0	7.87E-07	6.71E-28	0
Best	1.8	0	0	3.21E-07	1.45E-06	3.47E-18	0	2.13E-08	1.29E-28	0
Worst	1.8	0	0	7.7436531	9.85E-06	4.20E-16	0	4.93E-06	2.22E-27	0
Std	4.68E-16	0	0	2.3628901	2.83E-06	1.52E-16	0	1.52E-06	7.01E-28	0
F23 / Mean	-2.80E-25	-1.8013	-1.8013	-1.797961	-1.80127	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013
Best	-2.80E-25	-1.8013	-1.8013	-1.801302	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013
Worst	-2.80E-25	-1.8013	-1.8013	-1.785878	-1.80122	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013
Std	4.84E-41	0	0	0.0053225	2.87E-05	0	0	0	0	1.66E-16
F24 / Mean	-2.41E-17	-4.68765	-4.34262	-4.471775	-4.4394	-4.61012	-4.6493	-4.62716	-4.62595	-4.68766
Best	-2.41E-17	-4.68765	-4.6459	-4.644178	-4.56506	-4.68766	-4.68765	-4.68266	-4.68765	-4.68766
Worst	-2.41E-17	-4.68765	-3.56132	-4.092024	-4.35676	-4.49589	-4.49589	-4.51128	-4.49589	-4.68766
Std	6.50E-33	0	0.346957	0.1647279	0.068313	0.078629	0.071728	0.059187	0.082634	1.07E-15
F25 / Mean	-2.23E-11	-9.66014	-8.27455	-8.448917	-6.42466	-8.72957	-8.54008	-6.63433	-9.2148	-9.65598
Best	-2.23E-11	-9.66014	-9.22086	-9.087379	-6.75966	-9.21287	-9.46413	-7.54069	-9.54331	-9.66015
Worst	-2.23E-11	-9.66014	-7.30718	-7.628781	-5.95715	-7.82429	-6.06211	-5.6037	-8.68437	-9.61839
Std	1.19E-19	0	0.702673	0.5173604	0.281195	0.463703	0.979792	0.579317	0.248296	0.013207
F26 / Mean	9.352615	3.37E-12	80.29287	50.56716	159.9998	0	79.26627	6.06E-11	56.21504	0.650299
Best	6.964713	2.84E-13	15.91934	23.002178	128.1243	0	44.77347	0	39.79829	0
Worst	12.93447	1.44E-11	129.344	82.179468	185.2772	0	111.4745	2.70E-10	83.57621	6.502992
Std	1.945195	4.53E-12	30.34942	18.263706	17.77339	0	23.4953	9.88E-11	13.7805	2.056427

جدول ۹: نتایج ارزیابی آزمون دوگانه (Pairwise) برای نتایج توابع استاندارد

Method-1 vs Method-2	Statistic	Sig.
NFA-ACVO	.212	.801
NFA-FA	.558	.507
NFA-ABC	.942	.262
NFA-SO	.981	.243
NFA-WO	1.365	.104
NFA-PSO	2.308	.006
NFA-FFO	2.346	.005
NFA-CDO	3.000	.000
NFA-EVO	5.019	.000



شکل ۳: نمودار همگرایی بهترین جواب در CPU-Time برای توابع F26 و F25, F20, F19, F18, F13, F9, F4, F1



شکل ۴: نتایج ارزیابی آزمون Friedman برای میانگین نتایج بهینه سازی توابع استاندارد

برای بررسی برتری الگوریتم مطرح شده، این روش را علاوه بر روش پایه خود (یعنی FA) با سایر روش های فرا ابتکاری برگرفته از طبیعت که عملکرد بهتری نسبت به بقیه روش ها دارند مانند الگوریتم ازدحام ذرات (PSO)، الگوریتم زنبور عسل مصنوعی (ABC)، الگوریتم گرگ خاکستری (GWO) و الگوریتم بهینه سازی وال (WOA) مقایسه می کنیم. برای تنظیم پارامترها از جدول (۶) استفاده کرده ایم و نتایج آماری بدست آمده حاصل ۳۰ بار اجرا توسط الگوریتم ها می باشد. شرط پایان برای اجرا مدت ۱۰ ثانیه اجرا ($CPU - TIME = 10$) می باشد.

در ارزیابی مجموعه داده های ($Datasets$) مطرح شده، تعداد سرویس های موازی برای هر سرویس انتزاعی با عدد n مشخص شده است. تعداد سرویس انتزاعی m و تعداد سرویس موازی n برای هر سرویس انتزاعی، دو پارامتر اساسی برای مسئله ترکیب سرویس های ابری است. برای داده های QWS و تصادفی مقادیر m و n را تغییر می دهیم تا تاثیر آن ها را در کیفیت ترکیب سرویس و زمان محاسبات این الگوریتم بدست آوریم.

جدول ۱۰: بررسی تاثیر مقدار پارامتر θ در عملکرد الگوریتم NFA

θ	csQoS	STD	Best	Rank
0.1	0.7731	0.0029	0.8321	6
0.5	0.8108	0.0049	0.8503	5
0.9	0.8832	0.0021	0.9132	1
1.5	0.8621	0.0014	0.8712	2
5	0.8321	0.0043	0.8653	3
10	0.8245	0.0011	0.8521	4

جدول ۱۱: بررسی تاثیر مقدار پارامتر $P_{parasitism}$ در عملکرد الگوریتم NFA

$P_{parasitism}$	csQoS	STD	Best	Rank
0.1	0.8123	0.0029	0.8321	5
0.3	0.8432	0.0076	0.8635	3
0.5	0.8822	0.0034	0.8952	1
0.7	0.8211	0.0093	0.8321	4
0.9	0.8526	0.0029	0.8722	2
0	0.7643	0.0015	0.8111	6

۵-۱- کنترل تاثیر پارامترها در روش مطرح شده

جدول (۱۰) و (۱۱) برای کنترل تاثیر پارامترهای اساسی روش مطرح شده طراحی شده است. در این جدولها تاثیر پارامتر θ در جدول (۱۰) بررسی شده است. در این جدول مشخص است که مقدار $\theta = 0.9$ از سایر مقادیر دارای امتیاز بهتری بوده و برای ترکیب سرویس های ابری می توان از این مقدار پارامتر استفاده کرد. مقادیر بیشتر این پارامتر تاثیر منفی کمتری نسبت به مقادیر کمتر آن دارد. برای مثال برای مقدار $\theta = 0.1$ مقدار $csQoS$ برابر با 0.7731 است و این درحالی است که برای مقدار $\theta = 10$ این مقدار برابر با 0.8445 می باشد. در جدول (۱۱) تاثیر پارامتر $P_{parasitism}$ بر عملکرد NFA بررسی شده است. در این جدول مقدار 0.5 بهترین نتیجه را برای بدست آوردن هزینه ترکیب سرویس ابری دارد. مقادیر کمتر این پارامتر تاثیر منفی بیشتر نسبت به مقادیر بزرگتر دارد. در این جدولها STD میزان انحراف معیار، $Best$ بهترین $csQoS$ و $Rank$ امتیاز آن حالت از پارامتر می باشد.

در جدولهای (۱۲) و (۱۳) الگوریتم مطرح شده را با روش پایه FA مقایسه می کنیم تا بتوانیم میزان تاثیر تغییرات انجام شده برای ترکیب سرویس های ابری را ببینیم. در جدول (۱۲) میزان تاثیر سرویس های موازی بررسی شده است. برای اینکه تاثیر مقدار n در ترکیب سرویس های ابری بررسی شود مقدار آن را از ۵۰ تا ۵۰۰ تغییر داده ایم و برای هر حالت مقدار هزینه $csQoS$ را محاسبه کردیم. مشخص است که هر چقدر تعداد سرویس های موازی زیاد می شود، پیدا کردن جواب بهینه سخت می شود. $Time$ میزان زمانی را نشان می دهد تا به بهینه در مقدار تکرار الگوریتم 5000 به آن رسیده است. از جدول (۱۲) مشخص است که الگوریتم NFA نسبت به FA بهتر عمل کرده و توانسته است در مقادیر n مختلف جواب های بهینه تری را بدست آورد که دارای بالاترین سطح کیفیت در ترکیب سرویس ابری است که انجام داده است. در این جدولها از داده های QWS استفاده شده است.

جدول ۱۲: مقایسه FA با NFA در مقادیر مختلف n توسط داده های QWS

FA				
n	csQoS	Std	Time	Std2
50	0.8141	0.0012	10.1245	0.0163
100	0.8032	0.0031	11.3241	0.0142
200	0.7401	0.0042	12.2817	0.0521
300	0.6843	0.0058	16.7554	0.0143
400	0.6241	0.0079	19.2532	0.0832
500	0.5521	0.0071	20.5433	0.0321
NFA				
50	0.8723	0.0054	10.4326	0.0175
100	0.8532	0.0031	10.8432	0.0543
200	0.8211	0.0043	11.1454	0.0216
300	0.8109	0.0016	11.3456	0.0432
400	0.7421	0.0037	13.2965	0.0631
500	0.6821	0.0042	15.8765	0.0521

جدول ۱۳: مقایسه FA با NFA در مقادیر مختلف m توسط داده های QWS

FA				
m	csQoS	Std	Time	Std2
5	0.8334	0.0065	10.1546	0.0656
10	0.8145	0.0021	10.7899	0.0324
20	0.7277	0.0076	11.7654	0.0988
30	0.7011	0.0038	13.7678	0.0453
40	0.6532	0.0062	18.8766	0.0321
50	0.5845	0.0011	19.9787	0.0651
NFA				
5	0.8711	0.0055	10.4326	0.0765
10	0.8355	0.0027	11.8432	0.0432
20	0.8111	0.0097	12.9657	0.0785
30	0.8109	0.0015	13.8565	0.0357
40	0.7021	0.0042	17.1256	0.0213
50	0.6421	0.0076	18.5476	0.0247

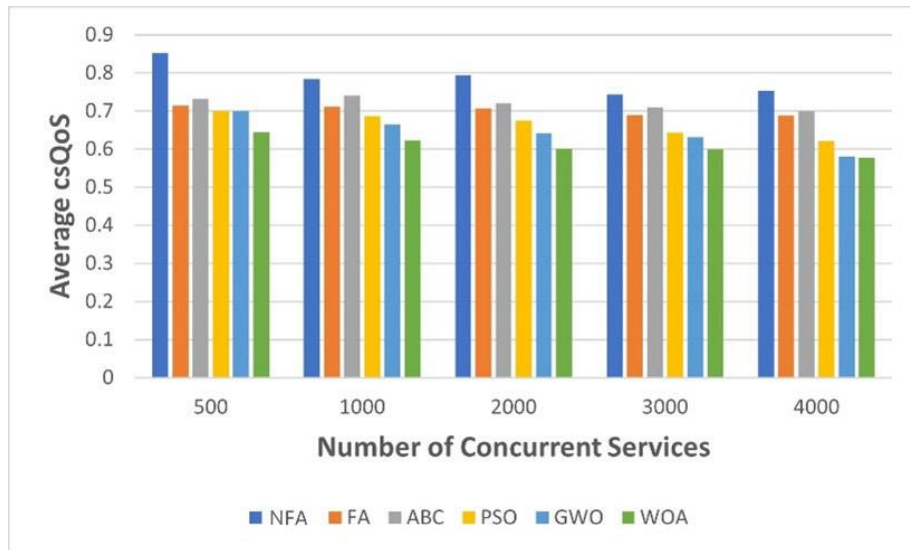
جدول (۱۳) برای مقادیر مختلف m همانند جدول (۱۲) بررسی شده است. مقادیر m از ۵ تا ۵۰ تغییر کرده است تا تاثیر تغییر تعداد سرویس های انتزاعی بر عملکرد الگوریتم بررسی شود. مشخص است که روش مطرح شده نسبت به FA بهتر عمل کرده و توانسته است مقادیر csQoS بالایی را بدست آورد که نشان از بالا بودن کیفیت ترکیب سرویس هایی است که بدست آورده است. شرایط ارزیابی برای این جدول همانند جدول (۱۲) می باشد.

جدول (۱۴) نتایج ارزیابی بر روی ترکیب سرویس های ابری داده های تصادفی با استفاده از الگوریتم های معروف GWO, WOA, ABC, PSO, FA و روش مطرح شده NFA را نشان می دهد. در این جدول بر روی مقادیر بسیار بالایی از m و n نتایج را بدست آورده ایم، شرط پایان جستجوی نتایج همانند جدول های قبلی می باشد. نتایج آماری حاصل ۳۰ بار اجرا برای هر

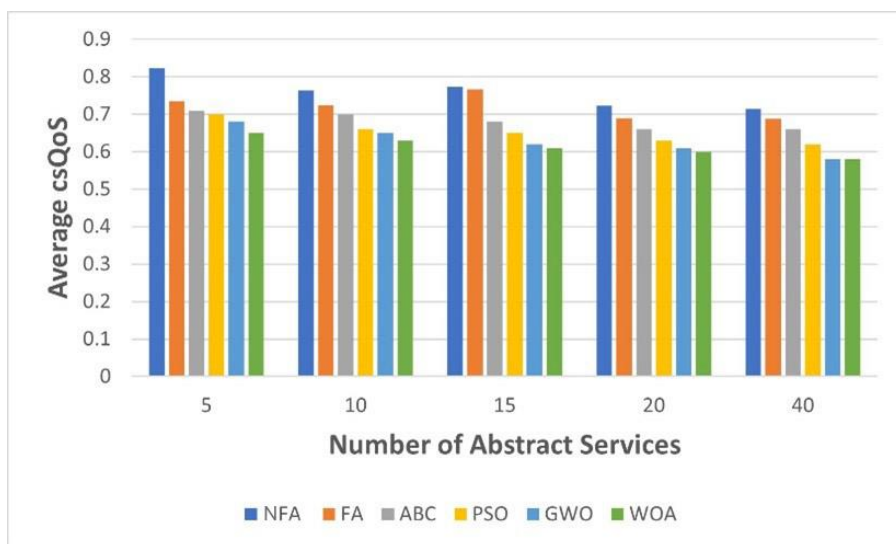
الگوریتم می‌باشد. همان طور که از جدول مشخص است روش مطرح شده توانسته است در هر حالت از مقادیر m و n میزان کیفیت سرویس بالایی را بدست آورد. این مقدار در جدول به صورت پررنگ تر مشخص شده است.

جدول ۱۴: مقایسه NFA در مقادیر مختلف n و m با سایر الگوریتم‌های فراابتکاری معروف

method	csQoS	Std	Time
m=100, n=10000			
GWO	0.4103	0.0034	25.1546
WOA	0.4453	0.0074	49.7899
ABC	0.6432	0.0026	11.7654
PSO	0.5456	0.0087	13.7678
FA	0.5893	0.0016	14.8766
NFA	0.6853	0.0039	12.9787
m=200, n=20000			
GWO	0.3854	0.0048	28.4326
WOA	0.4023	0.0036	56.8432
ABC	0.6322	0.0019	16.9657
PSO	0.5733	0.0036	23.8565
FA	0.6122	0.0092	14.1256
NFA	0.6643	0.0018	13.5476
m=4000, n=5000			
GWO	0.3398	0.0036	30.1245
WOA	0.4250	0.0027	20.3241
ABC	0.6192	0.0059	15.2817
PSO	0.5322	0.0035	19.7554
FA	0.6044	0.0096	10.2532
NFA	0.6344	0.0043	12.5433
m=2000, n=1000			
GWO	0.8711	0.0043	20.4326
WOA	0.8355	0.0065	41.8432
ABC	0.8111	0.0031	12.9657
PSO	0.8109	0.0084	17.8565
FA	0.7021	0.0039	14.1256
NFA	0.6421	0.0063	16.5476



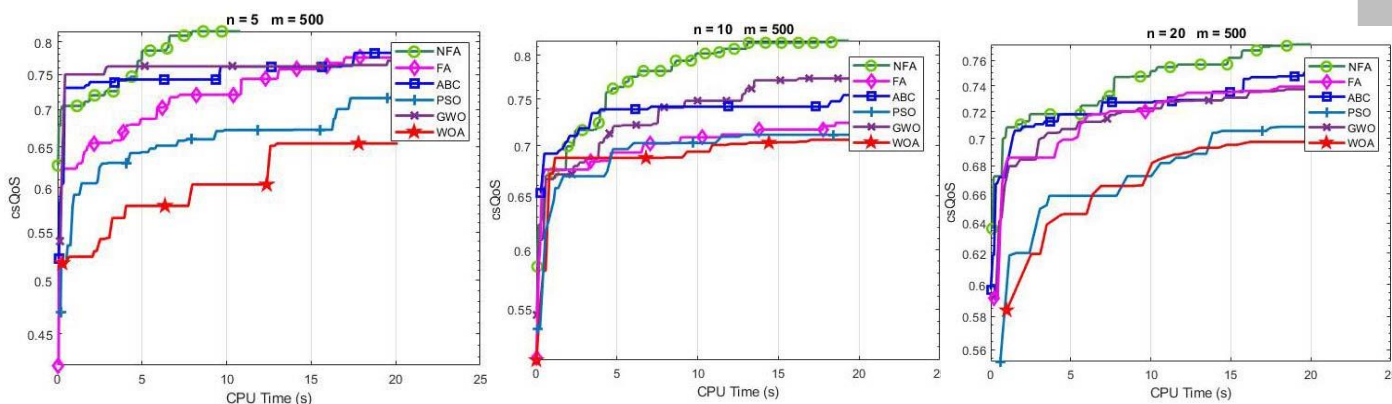
شکل ۵: نمودار میله‌ای تاثیر تعداد سرویس‌های انتزاعی m در ترکیب سرویس‌های ابری داده‌های QWS



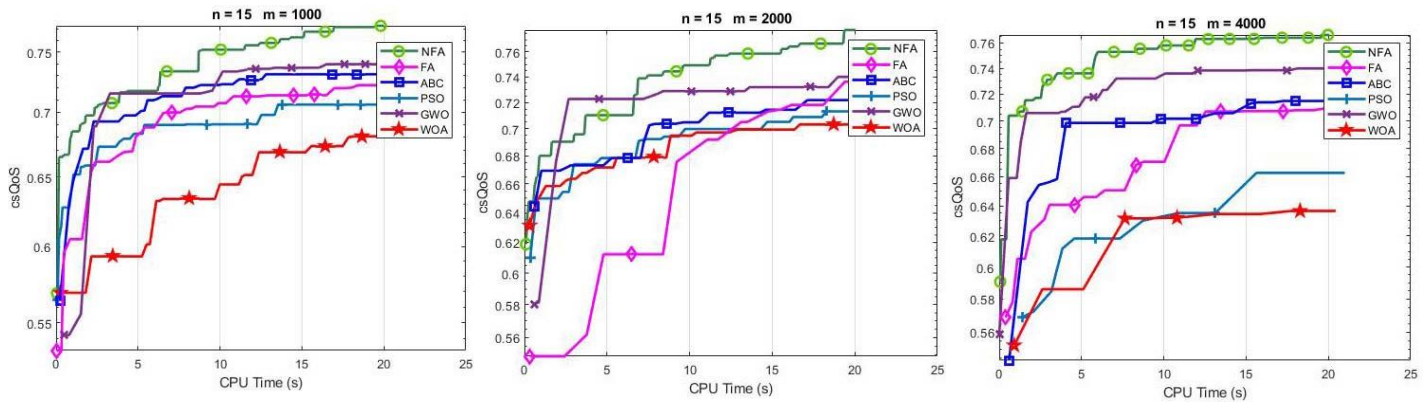
شکل ۶: نمودار میله ای تاثیر تعداد سرویس های موازی n در ترکیب سرویس ابری داده های QWS

شکل (۵) تاثیر تغییر سرویس های انتزاعی را در ترکیب سرویس های داده های QWS نمایش می دهد و شکل (۶) تاثیر تغییر سرویس های موازی را در ترکیب سرویس های این داده ها نمایش می دهد. همانطور که مشخص است در این شکل ها روش مطرح شده که به رنگ میله آبی می باشد در اکثر حالات بالاتر از همه الگوریتم های معروف دیگر می باشد که این نشان دهنده بدست آوردن کیفیت سرویس بالاتر نسبت به روش های دیگر است.

شکل (۷) و (۸) نمودارهای همگرایی را برای بدست آوردن بهترین ترکیب سرویس ابری در داده های تصادفی نشان می دهد که برای الگوریتم معروف مطرح شده بررسی شده است. در این نمودار ها روش مطرح شده توانسته است نسبت به بقیه همگرایی سریع و دقیقی را از خود نشان دهد. شکل (۷) برای مقادیر مختلف n و شکل (۸) برای مقادیر مختلف m بررسی شده است.



شکل ۷: نمودارهای همگرایی بررسی تاثیر n ، در ترکیب سرویس داده های QWS



شکل ۸: نمودارهای همگرایی بررسی تاثیر m ، در ترکیب سرویس داده‌های QWS

۶- نتیجه گیری

در الگوریتم کرم شب‌تاب که الگوریتمی برگرفته از طبیعت است، رفتارهای جدید زیستی کرم شب‌تاب، به طور کامل بررسی نشده است. در این مقاله این رفتارهای جدید که بر اساس آخرین رفتارهای کرم شب‌تاب یعنی حرکت به سمت جمعیت و حالت شکارچی کرم شب‌تاب می‌باشد بررسی شد و به الگوریتم کرم شب‌تاب اضافه گردید. با اعمال این تغییرات، دو پارامتر جدید θ و $P_{parasitism}$ معرفی شدند که این پارامترها به طور مستقل مورد بررسی قرار گرفتند. بررسی‌ها نشان داد که پارامتر $P_{parasitism}$ که باعث انتقال به حالت شکارچی می‌شود، در بهبود راه حل بسیار موثر است. عدم وجود آن در کاهش کارایی الگوریتم تاثیر گذار است و تغییرات اعمال شده در تابع بروزرسانی برای برتری روش پیشنهادی کافی نیست. هر دو این تغییرات، یعنی حالت شکارچی و حرکت به سمت جمعیت، به طور مستقل موثر نیستند و تنها زمانی که با هم اعمال شوند، نتایج قابل توجهی نشان می‌دهند. برای بررسی روش پیشنهادی، از ۲۶ تابع استاندارد و ترکیب سرویس‌های ابری داده‌ها استفاده کردیم. نتایج بهینه‌سازی ۲۶ تابع استاندارد نشان می‌دهد که روش پیشنهادی نه تنها بهتر از الگوریتم کرم شب‌تاب قبلی عمل می‌کند، بلکه با سایر الگوریتم‌های فرا ابتکاری جدید مانند الگوریتم WO ، FFO و $ACVO$ قابل رقابت می‌باشد. ارزیابی‌های مبتنی بر روش‌های بدون پارامتر، آزمون $Friedman$ و آزمون دوگانه این برتری را بیشتر ثابت کردند و همچنین نمودارهای همگرایی مبتنی بر CPU - $Time$ سرعت بهتر همگرایی روش را نشان دادند. ترکیب سرویس‌های ابری با روش پیشنهادی نتایج قابل توجه و قابل مقایسه ای را نشان داد، به طوری که از روش‌های مرسوم GWO و WOA بهتر عمل کرد و حتی نسبت به آنها همگرایی مناسبی را نشان داد. از آنجایی که الگوریتم FA به طور گسترده مورد استفاده قرار می‌گیرد، می‌توانیم از روش پیشنهادی به جای FA برای کارهای آینده استفاده کنیم.

- [1] Z. Wu, S. Deng, and J. Wu, "Chapter 7 - Service Composition," in *Service Computing*, Z. Wu, S. Deng, and J. Wu Eds. Boston: Academic Press, 2015, pp. 177-227.
- [2] E.-G. Talbi and G. El, *Metaheuristics: From Design to Implementation*. 2009.
- [3] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," Berlin, Heidelberg, 2009: Springer Berlin Heidelberg, in *Stochastic Algorithms: Foundations and Applications*, pp. 169-178 .
- [4] S. Lewis, *Silent Sparks: The Wondrous World of Fireflies*. Princeton University Press, 2016.
- [5] L. F. Faust, *Fireflies, Glow-worms, and Lightning Bugs: Identification and Natural History of the Fireflies of the Eastern and Central United States and Canada*. University of Georgia Press, 2017.
- [6] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, p. 6915, 2010. [Online]. Available: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm.
- [7] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb Eds .Boston, MA: Springer US, 2010, pp. 760-766.
- [8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014/03/01/ 2014, doi: <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [9] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016/05/01/ 2016, doi: <https://doi.org/10.1016/j.advengsoft.2016.01.008>.
- [10] M. Abdel-Basset, R. Mohamed, M. Zidan, M. Jameel, and M. Abouhawwash, "Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 415, p. 116200, 2023/10/01/ 2023, doi: <https://doi.org/10.1016/j.cma.2023.116>.
- [11] H. Zaher, H. Al-Wahsh, M. H. Eid, R. S. A. Gad, N. Abdel-Rahim, and I. M. Abdelqawee, "A novel harbor seal whiskers optimization algorithm," *Alexandria Engineering Journal*, vol. 80, pp. 88-109, 2023/10/01/ 2023, doi: <https://doi.org/10.1016/j.aej.2023.08.040>.

- [12] M. Han, Z. Du, K. F. Yuen, H. Zhu, Y. Li, and Q. Yuan, "Walrus optimizer: A novel nature-inspired metaheuristic algorithm," *Expert Systems with Applications*, vol. 239, p. 122413, 2024/04/01/ 2024, doi: <https://doi.org/10.1016/j.eswa.20>.
- [13] F. A. Hashim and A. G. Hussien, "Snake Optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, p. 108320, 2022/04/22/ 2022, doi: <https://doi.org/10.1016/j.knosys.2022.108320>.
- [14] H. Emami, "Anti-coronavirus optimization algorithm," *Soft Computing*, vol. 26, no. 11, pp. 4991-5023, 2022/06/01 2022, doi: 10.1007/s00500-022-06903-5.
- [15] H. A. Shehadeh, "Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization," *Neural Computing and Applications*, vol. 35, no. 15, pp. 10733-10749, 2023/05/01 2023, doi: 10.1007/s00521-023-08261-1.
- [16] M. Azizi, U. Aickelin, H. A. Khorshidi, and M. Baghalzadeh Shishehgarhaneh, "Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization," *Scientific Reports*, vol. 13, no. 1, p. 226, 2023/01/05 2023, doi: 10.1038/s41598-022-27344-y.
- [17] K. Zervoudakis and S. Tsafarakis, "A global optimizer inspired from the survival strategies of flying foxes," *Engineering with Computers*, vol. 39, no. 2, pp. 1583-1616, 2023/04/01 2023, doi: 10.1007/s00366-021-01554-w.
- [18] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355, 2006, doi: <https://doi.org/10.1016/j.ecoinf.2006.07.003>.
- [19] C. Jatoth, G. R. Gangadharan, and U. Fiore, "Optimal fitness aware cloud service composition using modified invasive weed optimization," *Swarm and Evolutionary Computation*, vol. 44, pp. 1073-1091, 2019/02/01/ 2019, doi: <https://doi.org/10.1016/j.swevo.2018.11.001>.
- [20] M. B. Karimi, A. Isazadeh, and A. M. Rahmani, "QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm," *The Journal of Supercomputing*, vol. 73, no. 4, pp. 1387-1415, 2017/04/01 2017, doi: 10.1007/s11227-016-1814-8.
- [21] J. Dogani and F. Khunjush, "Cloud Service Composition Using Genetic Algorithm and Particle Swarm Optimization," in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, 28-29 Oct. 2021 2021, pp. 98-104, doi: 10.1109/ICCKE54056.2021.9721500 .

- [22] H. Wang, Y. Ding, and H. Xu, "Particle swarm optimization service composition algorithm based on prior knowledge," *Journal of Intelligent Manufacturing*, 2022/10/01 2022, doi: 10.1007/s10845-022-02032-w.
- [23] N. Bacanin, M. Zivkovic, T. Bezdán, K. Venkatachalam, and M. Abouhawwash, "Modified firefly algorithm for workflow scheduling in cloud-edge environment," *Neural Computing and Applications*, vol. 34, no. 11, pp. 9043-9068, 2022/06/01 2022, doi: 10.1007/s00521-022-06925-y.
- [24] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting," *Expert Systems with Applications*, vol. 41, no. 13, pp. 6047-6056, 2014/10/01/ 2014, doi: <https://doi.org/10.1016/j.eswa.2014.03.053>.
- [25] D.-K. Bui, T. Nguyen, J.-S. Chou, H. Nguyen-Xuan, and T. D. Ngo, "A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete," *Construction and Building Materials*, vol. 180, pp. 320-333, 2018/08/20/ 2018, doi: <https://doi.org/10.1016/j.conbuildmat.2018.05.201>.
- [26] X.-S. Yang, "Nature-Inspired Optimization Algorithms," 2014.
- [27] A. O. Bush, J. C. Fernández, G. W. Esch, J. R. Seed, and J. C. F. Ndez, *Parasitism: The Diversity and Ecology of Animal Parasites*. Cambridge University Press, 2001.
- [28] A. Y. Alfakih, *Euclidean Distance Matrices and Their Applications in Rigidity Theory*. Springer International Publishing, 2018.
- [29] E. Al-Masri, and Mahmoud Q. H., "QoS-based Discovery and Ranking of Web Services," presented at the IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007.
- [30] G. Kang, B. Liang, L. Ding, J. Liu, B. Cao, and Y. Kang, "QoS-aware web service recommendation via exploring the users' personalized diversity preferences," *Engineering Reports*, vol. 6, no. 1, p. e12695, 2024, doi: <https://doi.org/10.1002/eng2.12695>.
- [31] W. Serrai, A. Abdelli, L. Mokdad, and Y. Hammal, "Towards an efficient and a more accurate web service selection using MCDM methods," *Journal of Computational Science*, vol. 22, pp. 253-267, 2017/09/01/ 2017, doi: <https://doi.org/10.1016/j.jocs.2017.05.024>.
- [32] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. Wiley, 2013.