دوره چهاردهم، شماره بهار و تابستان ۱۴۰۰

مجله فناوری اطلاعات در طراحی مهندسی

Information Technology in Engineering Design

http://ited.sinaweb.net

افزایش سرعت همگرایی الگوریتم کلونی زنبور عسل به کمک یادگیری تقویتی

آزاده جوادی [*(۱)] گلاره ویسی [(۲)]

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

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

(تاریخ ارسال: ۱۴۰۰/۰۸/۰۴ تاریخ پذیرش: ۱۴۰۰/۱۱/۰۵)

چکیده

الگوریتم کلونی زنبور عسل مصنوعی یک روش بهینه‌سازی تکاملی مبتنی بر هوش جمعی است. اگرچه این الگوریتم در بسیاری از مسائل بهینه‌سازی پیچیده نتایج مطلوبی را کسب کرده، اما دارای مشکلاتی مانند عدم قدرت کافی در استخراج جواب‌های نهایی می‌باشد، که باعث شده این الگوریتم نسبت به سایر الگوریتم‌های مشابه، سرعت همگرایی کمتری داشته باشد. جهت حل این مشکل از یادگیری تقویتی استفاده می‌کنیم که به‌وسیله آن می‌توان در هر موقعیت، استراتژی بهینه را برای هر زنبور عسل تعیین نمود. از این رو، در این مقاله یک الگوریتم ممتیک زنبور عسل ارائه می‌شود که در آن چند عبارت بروز رسانی پیشنهاد شده و هر یک از این عبارات به نفع اکتشاف یا استخراج الگوریتم هستند. سپس یادگیری تقویتی، سیاستی را ارائه می‌دهد که توسط آن، هر زنبور عسل می‌تواند در هر گام، بهترین عملکرد را از بین عملکردهای موجود انتخاب نماید. جهت بررسی نتایج روش پیشنهادی و مقایسه آن با سایر الگوریتم‌های مبتنی برهوش ازدحامی، از توابع معیار مختلفی استفاده شده و نتیجه شبیه‌سازی‌ها صحت و سرعت بالاتر روش پیشنهادی در مقایسه با سایر الگوریتم‌های مشابه را نشان می‌دهد.

واژه‌های کلیدی: بهینه‌سازی، الگوریتم‌های تکاملی، الگوریتم زنبور عسل مصنوعی، اکتشاف، استخراج، یادگیری تقویتی

*عهده دار مکاتبات:

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

تلفن: ۰۵۱۳۸۴۴۸۳۵۸ پست الکترونیکی: *azadehjavadi1010@gmail.com*

۱. مقدمه

در دو دهه گذشته روش‌های بهینه‌سازی تصادفی مبتنی بر جمعیت گسترش چشمگیری داشته‌اند به طوری که در بسیاری از مسائل بهینه‌سازی پیچیده مانند برنامه ریزی مسیر ربات [۱, ۲] شبکه‌های حسگر بیسیم [۳] کاربردهای مالی [۴, ۵] بخش‌بندی عکس [۶, ۷] و کاربردهای راداری [۸] مورد استفاده قرار گرفته اند. دلیل اقبال عمومی به این الگوریتم‌ها را می‌توان سادگی پیاده سازی، سرعت همگرایی بالا و عدم گرفتار شدن در جواب‌های بهینه محلی دانست . الگوریتم کلونی زنبور عسل مصنوعی[۱] (ABC) یک روش بهینه‌سازی تکاملی نسبتا جدید است که در سال ۲۰۰۵ با الهام گرفتن از نحوه جستجوی هوشمند زنبورهای عسل جهت یافتن غذا ارائه شد [۹]. اگرچه که این الگوریتم در بسیاری از مسائل بهینه‌سازی پیچیده نتایج مطلوبی را کسب کرده است [۱۰, ۱۱] اما از مشکلاتی نیز رنج می‌برد که از آن جمله می‌توان به عدم قدرت کافی در استخراج جواب‌های پیرامون جواب‌های موجود، اشاره کرد. به عبارت دیگر جستجوی ABC در اکتشاف جواب‌های دور افتاده به خوبی عمل می‌کند (جستجوی سراسری) ولی در استخراج جواب‌های پیرامون جواب‌های موجود (جستجو محلی) عملکرد ضعیفی دارد [۱۲]. همین مسئله سرعت همگرایی الگوریتم ABC را کاهش می‌دهد. دلیل این مشکل را می‌توان در عبارت بروز رسانی جواب‌ها در فاز استخراج دانست که در هر زمان جواب جدید تنها با تغییر تصادفی در یکی از ویژگی‌های جواب موجود تولید می‌شود. در این حالت هر فرزند (جواب جدید) بسیاری از ویژگی‌های والد (جواب قدیمی) را به ارث می‌برد که این مسئله سبب کاهش قدرت استخراج ABC و در نتیجه کاهش سرعت همگرایی این الگوریتم شده است. برای حل این مشکل راه حل‌های متعددی ارائه شده است که مهمترین آن‌ها استفاده از الگوریتم‌های بهینه‌سازی مبتنی بر ممتیک است [۱۳]. هرچند که الگوریتم‌های ممتیک در بعضی از کاربردها از عملکرد مطلوبی برخوردار هستند اما از یک چالش مهم رنج می‌برند. این چالش مهم عدم آگاهی الگوریتم از زمان‌بندی مناسب مراحل استخراج و اکتشاف است [۱۴].

برای حل مشکلات فوق در این مقاله یک الگوریتم ممتیک ABC ارائه شده است که در آن از تعدادی عبارت بروز رسانی اکتشافی و استخراجی استفاده می‌شود. در روش ارائه شده از یادگیری تقویتی[۲] (RL) برای یافتن سیاستی استفاده می‌شود که موعد مناسب جهت بهره بردن از هر یک از عبارات مذکور را تعیین می‌نماید. از این رو روش پیشنهادی را الگوریتم کلونی زنبور عسل مصنوعی بهبود یافته به کمک یادگیری تقویتی[۳] (IABCRL) نام نهاده ایم. در IABCRL چگونگی بروز رسانی هر جواب تحت کنترل RL قرار دارد، به عبارت دیگر RL تعیین می‌کند که در هر زمان، هر جواب به‌وسیله کدامیک از عبارات مذکور مورد بروز رسانی قرار گیرد. در ادامه این مقاله ابتدا به بررسی پیشینه روش‌هایی خواهیم پرداخت که به حل مشکل استخراج الگوریتم ABC پرداخته اند. در بخش سوم به ارائه روش پیشنهادی می پردازیم و از جهات مختلف این روش را مورد تحلیل قرار خواهیم داد. در ادامه و در بخش چهارم نیز به ارائه نتایج شبیه‌سازی‌ها می‌پردازیم که در آن‌ها عملکرد روش پیشنهادی با تعدادی روش مشابه در این حوزه مقایسه شده است. در نهایت نیز، در بخش پنجم به نتیجه گیری و ارائه پیشنهاداتی برای کارهای آینده خواهیم پرداخت.

---

۱. Artificial Bee Colony

۲. Reinforcement learning

۳. Improved Artificial Bee Colony algorithm by Reinforcement Learning (IABCRL)

۲. کارهای مرتبط

در این بخش ابتدا به بررسی پیشنه روش‌هایی خواهیم پرداخت که جهت بهبود عملکرد الگوریتم ABC ارائه شده اند. سپس به بررسی پیشینه الگوریتم‌های تکاملی خواهیم پرداخت که در آن‌ها از RL جهت بهبود عملکردشان استفاده شده است.

۲-۱. الگوریتم‌های زنبورعسل بهبود یافته

روش‌هایی که جهت بهبود الگوریتم ABC ارائه شده‌اند را می‌توان به دو دسته تقسیم کرد: دسته اول روش‌هایی هستند که در حقیقت توسعه مستقیم ABC می‌باشند، به طوری‌که در آن‌ها صرفا نحوه ارث‌بری هر جواب جدید از جواب‌های والد تغییر کرده است. دسته دوم، روش‌هایی هستند که از ترکیب الگوریتم ABC با سایر الگوریتم‌های تکاملی بدست می‌آیند [۳۱-۳۵]. از جمله روش‌های دسته اول می‌توان به روش مقاله [۱۵] اشاره کرد که در آن یک پارامتر کنترلی معرفی شده، که جهت تعیین تعداد متغیرهای هر جواب که باید تغییر کنند مورد استفاده قرار می‌گیرد. همچنین در این مقاله از روش نگاشت آشوب [۱۶] جهت مقدار دهی اولیه به جمعیت جواب‌ها استفاده شده است. با استفاده از این روش درجه تصادفی بودن جواب‌های اولیه افزایش می‌یابد که در نتیجه جواب بهینه با سرعت بیشتری پیدا خواهد شد. در مقاله [۱۷] پارامتر کنترل دیگری به نام نرخ بهبود [۱] (MR) معرفی شده است که به وسیله آن درصدی از جواب‌های جدید که از جواب‌های قدیمی ارث می‌برند، مشخص می‌شود. در این مقاله برای هر ویژگی از هر جواب یک عدد تصادفی تولید می‌شود، در صورتی که این عدد تصادفی کمتر از پارامتر MR باشد، ویژگی مرتبط به صورت تصادفی و با استفاده از روش سنتی ABC تکامل می یابد، در غیر این صورت این ویژگی تغییر نمی‌کند. در مقاله [۱۳] نیز از تکنیک مقاله قبل جهت تعیین ویژگی‌هایی که نیاز به تکامل دارند استفاده شده، با این تفاوت که به جواب‌های با ارزیابی بسیار کم نیز تا حدی امکان توسعه داده شود که در تکامل‌های آینده بهجواب‌های مناسبی تبدیل شوند. در مقاله [۱۳] یک روش انتخاب مبتنی بر رتبه بندی تطبیقی به الگوریتم ABC اضافه شده است، در این روش احتمال انتخاب جواب‌های جدید (منابع غذایی مورد جستجو زنبورهای عسل و والدین) همه بر اساس رتبه‌بندی انتخاب می‌شوند و این رتبه‌بندی با توجه به وضعیت تکامل جمعیت تنظیم می‌شود. از جمله روش‌های دسته دوم می‌توان به روش مقاله [۱۹] با عنوان HDABCA اشاره کرد که در آن در هر گام، ابتدا جواب‌ها به‌وسیله الگوریتم ABC بروزرسانی می‌شوند، یعنی مراحل زنبورهای کارگر، ناظر و پیش آهنگ به طور کامل اجرا می‌شود، سپس الگوریتم DE [۲] جهت بهبود جواب‌های خوب حاصل از ABC به کار می‌رود. در مقاله [۲۰] نیز روشی مشابه مقاله قبل ارائه شده است که در آن الگوریتم‌های ABC و DE به صورت تکراری اجرا می‌شوند. علاوه بر این تلفیق، جهت افزایش قدرت همگرایی، عبارت بروز رسانی ABC با عبارت بروز رسانی که در مقاله [۲۱] ارائه شده است به صورت تصادفی تلفیق شده است. در مقاله [۲۲] الگوریتم ABC با DE به شکلی متفاوت از دو مقاله قبلی تلفیق شده است. در این مقاله عبارات بروز رسانی DE به همراه ABC در فاز زنبورهای کارگر تلفیق می‌شوند. به عبارت دیگر در این فاز ابتدا یک عدد تصادفی تولید شده، اگر این عدد تصادفی از یک پارامتر وفقی کوچکتر باشد، عبارت بروز رسانی ABC اجرا می‌شود، در غیر این صورت بروز رسانی به روش DE انجام خواهد گرفت. در مقاله [۲۳] تلفیق ABC با DE در فاز زنبورهای ناظر صورت می‌گیرد و بروز رسانی زنبورهای کارگر با استفاده از یک عبارت بهبودیافته ABC صورت می‌پذیرد. برای رفع مشکلات روش‌های دسته دوم راه حل های متعددی ارائه شده است که مهمترین آن‌ها استفاده از الگوریتم‌های بهینه‌سازی مبتنی بر ممتیک است. در روش‌های ممتیک دو عبارت جستجوی محلی و جستجوی سراسری با یکدیگر ترکیب می‌شوند، به عبارت دیگر عبارت بروزرسانی الگوریتم‌های ممتیک شامل استخراج و اکتشاف می‌باشد. در مقاله [۲۴] دو استراتژی جستجوی پیشنهاد می شود که یکی دارای قابلیت اکتشاف قوی و دیگری دارای قابلیت استخراج قوی است؛ براساس سازگاری دو استراتژی جستجو

1. Artificial Bee Colony
2. Reinforcement learning
3. Improved Artificial Bee Colony algorithm by Reinforcement Learning (IABCRL)
4. Modification rate
5. Differential Evolution (DE) algorithm

با حل مساله و فرآیند جستجو، احتمال انتخاب هر استراتژی جستجو به صورت پویا با توجه به نرخ موفقیت تنظیم می‌شود و سپس بهینه‌سازی مشارکتی دو استراتژی جستجو برای بهبود عملکرد الگوریتم محقق می‌شود.

۲-۲. بهبود روش‌های بهینه‌سازی مبتنی بر هوش ازدحامی با استفاده از یادگیری تقویتی

روش‌های دیگری که لازم است در این بخش مورد بررسی قرار گیرد، روش‌هایی هستند که در آن‌ها از یادگیری تقویتی جهت تعیین عبارت بروز رسانی استفاده می‌شود. در اولین تلاش در این حوزه، مقاله [۲۵] از ترکیب الگوریتم بهینه‌سازی انبوه ذرات[1] به همراه یاگیری تقویتی جهت یافتن جواب در محیط‌های نویزی استفاده کرده است. در این مقاله از یادگیری تقویتی در فاز انتخاب جواب جدید استفاده شده است. در مقاله [۱] نیز از RL جهت تنظیم پارامترهای الگوریتم DE بهره برده شده و از روش حاصل جهت بهینه‌سازی مسیر ربات استفاده شده است. مهمترین تلاشی که در این حوزه انجام شده مربوط به مقاله [26] است که از RL جهت تلفیق هوشمند دو فاز استخراج و اکتشاف استفاده شده است. در این مقاله، رویکرد استفاده از RL جهت بهبود عملکرد الگوریتم PSO بکار برده شده است. در روش ارائه شده، عبارت بروز رسانی PSO با پارامترهای متفاوت جهت بروز رسانی یک ذره در نظر گرفته شده است و این مسئله که در هر زمان از چه پارامتری برای هر ذره استفاده شود، توسط RL تعیین می‌گردد. در این مقاله با در نظر گرفتن پارامترهای متفاوت سعی شده است عبارات بروز رسانی با خصوصیات استخراج و اکتشاف متفاوت ایجاد شود.

۳. روش پیشنهادی (IABCRL)

در این بخش به معرفی الگوریتم تکاملی خواهیم پرداخت که در آن جهت بهبود سرعت همگرایی ABC از روش یادگیری تقویتی استفاده شده است. در روش پیشنهادی، به جای استفاده از یک عملیات بروز رسانی، چندین عملیات بروز رسانی وجود دارد که در هر موقعیت و برای هر ذره یکی از این عملیات‌ها توسط یادگیری تقویتی انتخاب می‌شود. عملیات مختلف بروز رسانی به گونه ای تعیین شده اند که یه‌وسیله آن‌ها بتوان به ویژگی‌های مثبت و مختلف یک الگوریتم تکاملی از جمله اکتشاف، استخراج و همگرایی سریع دست یافت.

در ادامه ابتدا به بیان دلایل استفاده از یادگیری تقویتی جهت بهبود عملکرد ABC خواهیم پرداخت، سپس به معرفی عبارات بروز رسانی منتخب می‌پردازیم و علت انتخاب هر یک از آنها بیان خواهد شد. پس از آن، به بیان مراحل مختلف الگوریتم تکاملی پیشنهادی خواهیم پرداخت و در پایان در مورد همگرایی الگوریتم پیشنهادی به جواب بهینه بحث خواهد شد.

۳-۱. کاربرد یادگیری تقویتی در بهبود عملکرد ABC

در مسئله یادگیری تقویتی، یک عامل یادگیر با توانایی انجام حرکات مشخص (A) در یک محیط تعریف شده، قرار دارد. این عامل به ازاء حالتی که در آن قرار دارد ($s \in S$)، حرکتی را انجام می‌دهد ($a \in A$) و در مقابل انجام این حرکت پاداشی را دریافت می‌کند ($r \in R$) و به یک حالت جدید ($s' \in S$) (که می‌تواند همان حالت قبلی باشد) منتقل می‌شود. به تابعی که در حالت‌های مختلف حرکات مناسب را پیشنهاد می‌دهد، تابع استراتژی یا سیاست (Q) گفته می‌شود. این تابع به صورت زیر تعریف می‌شود:

$$Q: S \times A \rightarrow R \qquad (۱)$$

در صورتی که حالت‌ها (S) و حرکات (A) گسسته باشند، می‌توان تابع Q را به صورت یک ماتریس نشان داد که در این حالت ماتریس Q به وسیله الگوریتم Q_Learning قابل تعیین است.

در الگوریتم‌های تکاملی از جمله ABC در صورتی که عبارات بروز رسانی متعددی وجود داشته باشند، هر ذره را می‌توان یک عامل یادگیر در نظر گرفت که در محیط مسئله به دنبال کمینه کردن میزان تابع هدف خود است. برای هر ذره می‌توان حرکات (عبارات بروز رسانی) مختلفی طراحی کرد به طوری که هر کدام دارای خاصیت‌های متمایز هستند، به طوری که هر یک از آن‌ها تامین کننده یک یا چند

1. Particle Swarm Optimization (PSO)

ویژگی مورد نیاز الگوریتم تکاملی می‌باشند. از جمله این ویژگی‌ها می‌توان به استخراج (Ex)، همگرایی (C)، تلفیقی از استخراج و همگرایی (ExC) اشاره کرد. همچنین عبارت بروز رسانی ABC نیز به دلیل عملکرد خوب در بسیاری از کاربردها به عنوان یکی از حرکات در نظر گرفته خواهد شد. نکته بعدی که جهت استفاده یادگیری تقویتی در الگوریتم ABC باید تعیین شود حالت‌هایی است که ذره در آن قرار می‌گیرد. برای تعیین مجموعه حالات می‌توان برای هر حرکت یک حالت در نظر گرفته شود. در این صورت اگر چهار حرکت تعریف شود، چهار حالت نیز برای هر ذره وجود خواهد داشت که با انجام هر یک از این حرکت‌ها ذره به حالت متناظر با آن منتقل می‌شود. به عبارت دیگر اگر یک ذره، عمل استخراج را انتخاب کرده است، در حالت استخراج قرار می‌گیرد.

همچنین می‌توان، پاداشی که هر ذره بعد از انجام یک حرکت دریافت می‌کند را وابسته به چگونگی تغییر تابع هدف به ازاء آن ذره بعد از انجام حرکت در نظر گرفت. به عنوان مثال پاداش هر حرکت یک ذره به این صورت تعیین می‌شود که اگر تابع هدف ذره مورد نظر بعد از انجام حرکت کاهش پیدا کند، پاداشی برابر +۱ به ذره داده می‌شود، ولی اگر میزان تابع هدف تغییر نکرد، میزان پاداش ۰ و اگر تابع هدف افزایش یافت، پاداش −۱ به ذره داده خواهد شد. (به عبارت دیگر ذره جریمه می‌شود.) توجه شود این نحوه پاداش دهی برای مسائل کمینه سازی تعیین شده است و اگر مسئله مورد نظر بیشینه سازی بود باید معکوس پاداش‌های فوق در نظر گرفته شود. با توجه به شرایط ذکر شده برای هر ذره مجموعه‌های S ،A و R به صورت زیر خواهند بود:

$$A = S = \{Ex, C, ExC, ABC\}. \quad (۲)$$
$$R = \{-1, 0, 1\}$$

از آنجا که تعداد حرکات عامل و حالت‌های سیستم با یکدیگر برابرند بنابران ماتریس Q به صورت مربعی تشکیل خواهد شد. مهمترین مسئله در روش پیشنهادی تعیین مجموعه حرکت‌های ذره‌ها است که در قسمت بعدی تعیین خواهند شد.

۳-۲. انتخاب مجموعه حرکات

عبارات بروز رسانی که به عنوان حرکات یک ذره در نظر گرفته می‌شوند، به صورت زیر می‌باشند:

عبارت اول (Ex): این عبارت جهت کسب ویژگی استخراج به صورت زیر در نظر تعیین شده است:

$$v_i = x_{r_1} + F(x_{r_2} - x_{r_3}) \quad (۳)$$

که در آن r_1، r_2 و r_3 اندیس‌های تصادفی غیر یکسان و مخالف i از میان جمعیت جواب‌ها هستند و F یک مقدار تصادفی در بازه [0,1] می‌باشد. عبارت فوق همان عملیات DE/rand/1 است [۲۱] که به دلیل قدرت بالای استخراج آن انتخاب شده است.

عبارت دوم (ExC): این عبارت برای داشتن تلفیقی از ویژگی استخراج و همگرایی به بهترین جواب یافت شده، انتخاب شده است که به صورت زیر تعریف می‌شود:

$$v_{i.j} = x_{i.j} + \phi_{i.j}(x_{i.j} - x_{k.j}) + \varphi_{i.j}(x_{best.j} - x_{k.j}) \quad (۴)$$

که در آن k یک اندیس تصادفی غیر یکسان با i از میان جمعیت جواب‌ها است و $\phi_{i.j}$ و $\varphi_{i.j}$ دو عدد تصادفی هستند که به ترتیب در بازه‌های [0,1] و [0,2] انتخاب می‌شوند. این عبارت از مقاله [۲۵] به دلیل قدرت استخراج و همگرایی آن انتخاب شده است .

عبارت سوم (C): این عبارت جهت کسب ویژگی همگرایی به صورت زیر تعریف شده است:

$$v_{i.j} = x_{i.j} + \varphi_{i.j}(x_{best.j} - x_{k.j}) \quad (۵)$$

که در آن k و $\varphi_{i.j}$ همان تعاریف عبارت قبل را دارند.

عبارت چهارم (ABC): عبارت بروزرسانی ABC نیز به دلیل عملکرد خوب آن در بسیاری از کاربردها به عنوان یکی از گزینه‌ها درنظر گرفته می‌شود. این عبارت به صورت زیر تعریف می‌شود:

$$v_{i.j} = x_{i.j} + \varphi_{i.j}(x_{i.j} - x_{k.j}) \quad (۶)$$

که در آن $\varphi_{i.j}$ یک عدد تصادفی در بازه [-1,1] است و k همان تعریف قبلی را دارد.

۳-۳. وجه اشتراک و تمایز میان روش پیشنهادی IABCRL و ABC

ساختار کلی روش پیشنهادی IABCRL مانند ساختار ABC است که در آن جستجو جواب‌ها از نحوه جستجو منابع غذایی توسط کلونی زنبورها الهام گرفته است. به عبارت دیگر جستجو از دو فاز تشکیل شده است. در فاز اول زنبورهای کارگر به جستجوی تصادفی و هوشمند جواب‌های جدید می‌پردازند و در صورت یافتن جواب‌های بهتر آن‌ها را به جای جواب‌های قبلی جایگزین می‌کنند. در فاز دوم زنبورهای ناظر به توسعه جواب‌های یافت شده توسط زنبورهای کارگر می‌پردازند به طوری که جواب‌های بهتر احتمال توسعه بیشتری دارند. همچنین جهت افزایش قدرت اکتشاف الگوریتم، در صورتی که جوابی طی چند مرحله تکامل بهبود نیافت آن را با یک جواب تصادفی جایگزین می‌کنیم. از طرفی در الگوریتم زنبور عسل بروز رسانی جواب‌ها چه در فاز زنبورهای کارگر و چه در فاز زنبورهای ناظر به وسیله عبارت (۶) انجام می‌شود. اما با توجه به دلیل مطرح شده در ابتدای این بخش، این عبارت نمی‌تواند نیاز الگوریتم به ویژگی‌های استخراج و اکتشاف را به طور کامل بر طرف کند. از این رو در روش پیشنهادی IABCRL در فاز زنبورهای کارگر بروز رسانی به وسیله عبارات EX، EXC، C و ABC که در بخش قبل ارائه شده اند انجام می‌شود. تعیین این مسئله که برای هر ذره‌ها در زمان‌های مختلف از چه عبارت بروز رسانی استفاده کنند به وسیله RL صورت می‌گیرد. در قسمت بعد چگونگی کاربرد RL در مسئله فوق بیان خواهد شد.

۳-۴. کاربرد یادگیری تقویتی در تعیین عبارت بروز رسانی

در روش پیشنهادی برای هر ذره X_i یک ماتریس استراتژی به نام Q_iتعریف می‌شود بطوری‌که این ماتریس در ابتدا شروع به کار الگوریتم برای تمام ذره‌ها با صفر مقدار دهی شده است. به عبارت دیگر:

$$Q_i^t = 0_{n\times m}.\, i = \{1.2.\ldots..NPop\}.t = 0 \qquad (۷)$$

که در آن $NPop$ تعداد جمعیت جواب‌ها، n تعداد حالت‌ها و m نشان دهنده تعداد عبارات بروز رسانی است. از آنجا که در روش پیشنهادی به ازاء هر عبارت بروز رسانی یک حالت تعریف شده است بنابراین پارامترهای m و n با یکدیگر برابر خواهند بود. در شروع به کار الگوریتم، هر ذره به صورت تصادفی یکی از عبارت بروز رسانی (۳) تا (۶) را انتخاب کرده و به وسیله آن جواب جدیدی مانند V_i را تولید می‌کند. به عبارت دیگر:

$$V_i = E_i^t(X_i). \qquad (۸)$$
$$E_i^t \in \{Ex.ExC.C.ABC\}$$

که در آن E_i^t عبارت بروز رسانی است که برای ذره iام در زمان t انتخاب شده است. در صورتی که V_i نسبت به X_i بهتر ارزیابی شد پاداش $+1$ به ذره داده شده و X_i با V_i جایگزین می‌شود و در غیر این صورت جریمه -1 برای ذره در نظر گرفته می‌شود. همچنین حالت ذره با توجه به عبارت بروز رسانی انتخاب شده، تغییر می‌یابد. پس از انجام مراحل فوق، ماتریس استراتژی مربوط به ذره iام به صورت زیر به روز رسانی می‌شود:

$$\mathrm{Q_i^{t+1}(s_i^t.E_i^t)} = Q_i^t(s_i^t.E_i^t) + \alpha[r_{t+1} + \gamma max_a Q_i^t(s_i^{t+1}.a) - Q_i^t(s_i^t.E_i^t)] \qquad (۹)$$

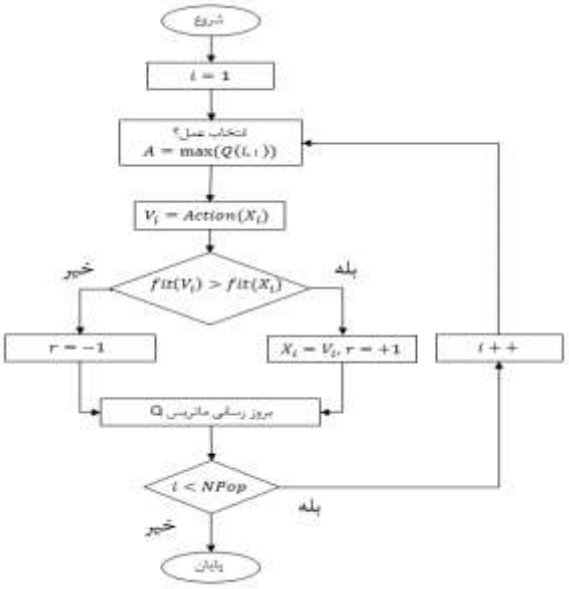
که در آن $r_{t+1} \in \{-1.+1\}$ پاداشی است که ذره به ازاء انجام حرکت E_i^t دریافت می‌کند. همچنین s_i^t و s_i^{t+1} به ترتیب حالات ذره iام در زمان tام و زمان $t+1$ را نشان می‌دهند. با توجه به اینکه در روش پشنهادی هر ذره با انتخاب هر عبارت، به حالت متناظر با آن عبارت منتقل می‌شود بنابراین:

$$s_i^{t+1} = E_i^t \qquad (۱۰)$$

در گام‌های بعدی (دوم الی آخر) عبارت بروز رسانی برای هر ذره با توجه به ماتریس استراتژی مربوط به خود انتخاب می‌شود. در این روش برای ذره X_i عبارتی انتخاب می‌شود که با توجه به حالت کنونی ذره بیشترین مقدار را در ماتریس Q_i^t داشته باشد. به عبارت دیگر:

$$E_i^t = arcmax_a Q_i^t(s_i^t.a) \quad (۱۱)$$

که در آن s_i^t حالتی است که ذره X_i پیش از انجام بروز رسانی در آن قرار دارد. مراحل فوق برای تمامی ذره‌ها انجام می‌شود. با توجه به مطالب ذکر شده می‌توان نمودار جریان مربوط به بروز رسانی جواب‌ها در فاز زنبورهای کارگر را به صورت شکل ۱ نشان داد:



شکل ۱. نمودار جریان روش پیشنهادی

۴. شبیه‌سازی‌ها و نتایج

در این بخش به مقایسه کیفیت روش پشنهادی (IABC_RL) با دیگر الگوریتم‌های بهینه‌سازی تکاملی مانند ABC [۲۵]، ABAC_ADE [۲۱]، CB_ABC [۱۳] خواهیم پرداخت. روش‌های مذکور از دو جنبه جهت مقایسه با روش پیشنهادی انتخاب شده اند. جنبه نخست شباهت آنها با ساختار ABC متداول است که دو روش ABC و CB_ABC به این دلیل انتخاب شده اند. جنبه دوم نیز استفاده تلفیقی از چند عملیات بروزرسانی به جای یک عملیات است که روش ABC_ADE از این جهت با روش پیشنهادی شباهت دارد.

۴-۱. توابع معیار

جهت مقایسه روش پیشنهادی با روش‌های فوق از تعداد توابع معیار استفاده شده است که در چهار نوع تک مد (Unimodal)، چند مد (Multimodal)، تفکیک پذیر (Separable) و تفکیک ناپذیر (Inseparable) تقسیم‌بندی می‌شوند. منظور از توابع تک مد توابعی هستند که تنها دارای یک بهینه محلی می‌باشد، به طوری که این بهینه محلی بر بهینه سراسری منطبق است. ویژگی استخراج یک الگوریتم بهینه‌سازی را می‌توان با توابع معیار تک مد ارزیابی کرد. اگر یک تابع معیار بیش از یک بهینه محلی داشته باشد، چنین تابعی را چند مد گویند. در این توابع ممکن است یک یا چند بهینه سراسری وجود داشته باشد. علاوه بر توانایی استخراج، توانایی اکتشاف یک الگوریتم بهینه‌سازی نیز می‌تواند به وسیله توابع معیار چند مد مورد سنجش قرار گیرند [۲۶]. تابع معیاری که بتوان آن را به وسیله جمع چند تابع معیار بازنویسی کرد، را تابع معیار تفکیک پذیر گویند و در مقابل توابع معیاری که به علت وابستگی بین متغیرهایشان، نتوان آنها را به‌وسیله جمع چند عبارت بازنویسی کرد، توابع معیار تفکیک ناپذیر گویند. در این توابع اگر یک متغیر تغییر کند، چندین متغیر دیگر تابع نیز دچار تغییر می‌شوند، بنابراین یافتن

جواب بهینه برای توابع معیار تفکیک ناپذیر بسیار پیچیده تر از توابع معیار تفکیک پذیر است [۲۶]. چنین توابعی در بسیاری از کاربردهای عملی دیده می‌شوند که از آن جمله می‌توان به شبکه‌های حسگر [۲۷] و سیستم‌های کنترل فازی [۲۸] اشاره کرد. در جدول ۱ مشخصات و نوع توابع معیار مورد استفاده بیان شده است.

جدول ۱. توابع معیار جهت مقایسه روش پیشنهادی با سایر روش‌ها

نام تابع	تابع	نوع	محدوده جستجو	کمینه
Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	US	$[-100 \quad 100]^D$	0
SumSquare	$f_2(x)=\sum_{i=1}^{D} i x_i^2$	US	$[-10 \quad 10]^D$	0
SumPower	$f_3(x)=\sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	MS	$[-10 \quad 10]^D$	0
Schwefel 2.22	$f_4(x)=\sum_{i=1}^{D} \lvert x_i\rvert+\prod_{i=1}^{D} \lvert x_i\rvert$	UI	$[-10 \quad 10]^D$	0
Step	$f_5(x)=\sum_{i=1}^{D} (\lfloor x_i+0.5\rfloor)^2$	US	$[-100 \quad 100]^D$	0
Exponential	$f_6(x)=\exp(0.5*\sum_{i=1}^{D} x_i^2)$	US	$[-1.28 \quad 1.28]$	0
Quartic	$f_7(x)=\sum_{i=1}^{D} i x_i^4+rand(0.1)$	US	$[-1.28 \quad 1.28]$	0
Griewank	$f_8(x)=\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	MI	$[-600 \quad 600]^D$	0

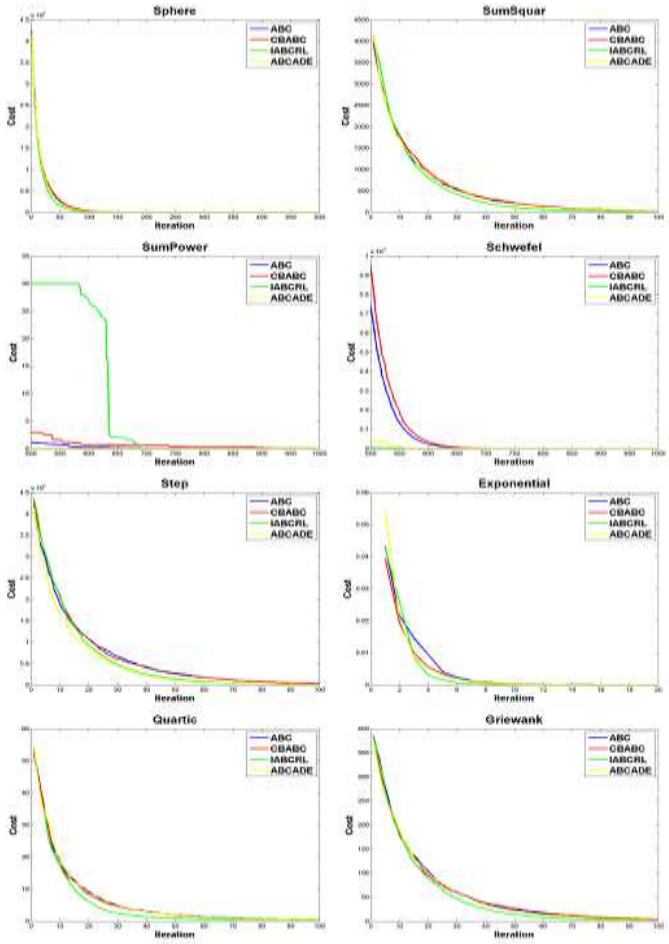
۴-۲. تنظیم پارامتر

برای اینکه بتوان مقایسه عادلانه‌ای بین روش پیشنهادی و روش‌های مذکور انجام داد، پارامترهای مشترک در تمامی روش‌های مذکور به صورت یکسان تنظیم می‌شوند. این پارامترهای مشترک شامل جمعیت جواب‌ها (SN)، ابعاد توابع معیار (D) و تعداد گام‌های الگوریتم می‌باشند. از این رو در تمامی آزمایش‌ها، جمعیت جواب در هر گام برابر ۱۰ در نظر گرفته شده است. ابعاد توابع معیار (D) در سه آزمایش مختلف برابر ۲۰، ۳۰ و ۵۰ در نظر گرفته شده است. همچنین جهت کاهش تاثیر مقدار دهی اولیه تصادفی جواب‌ها، تمامی آزمایش‌ها را ۵۰ مرتبه تکرار کرده و میانگین این تکرارها را به عنوان نتیجه کلی در نظر خواهیم گرفت.

۴-۳. آزمایش اول

در این آزمایش برای تمامی توابع معیار جدول ۱، متغیر D=20 در نظر گرفته شده است. همچنین برای هر روش تعداد جمعیت جواب‌ها ۱۰ و تعداد گام‌ها ۲۰۰۰ در نظر گرفته شده است. در جدول ۲ میانگین و انحراف معیار جواب‌های یافت شده آزمایش اول بعد از همگرایی برای الگوریتم پیشنهادی و الگوریتم‌های مشابه مذکور نشان داده شده است. همانطور که در جدول ۲ قابل مشاهده است، روش پیشنهادی در بهینه‌سازی توابع Sphere، Quartic و Griewank هم از نظر میانگین و هم از نظر انحراف معیار بهتر از سایر روش‌ها عمل کرده است. همچنین روش پیشنهادی در بهینه‌سازی توابع SumSquare از نظر میانگین و در بهینه‌سازی Exponential از نظر انحراف معیار بهتر از سایر روشهای عمل کرده است. در شکل ۲ نیز روند همگرایی روش پیشنهادی و سایر روش‌های مشابه نشان داده شده است. همانطور که در شکل ۲ قابل مشاهده است، روش پیشنهادی در ۱۰۰۰

تکرار اول در بهینه‌سازی توابع Sphere، SumSquare، Schwefel، Exponential، Quartic و Griewank زودتر از سایر روش‌ها به جواب بهینه همگرا شده است.



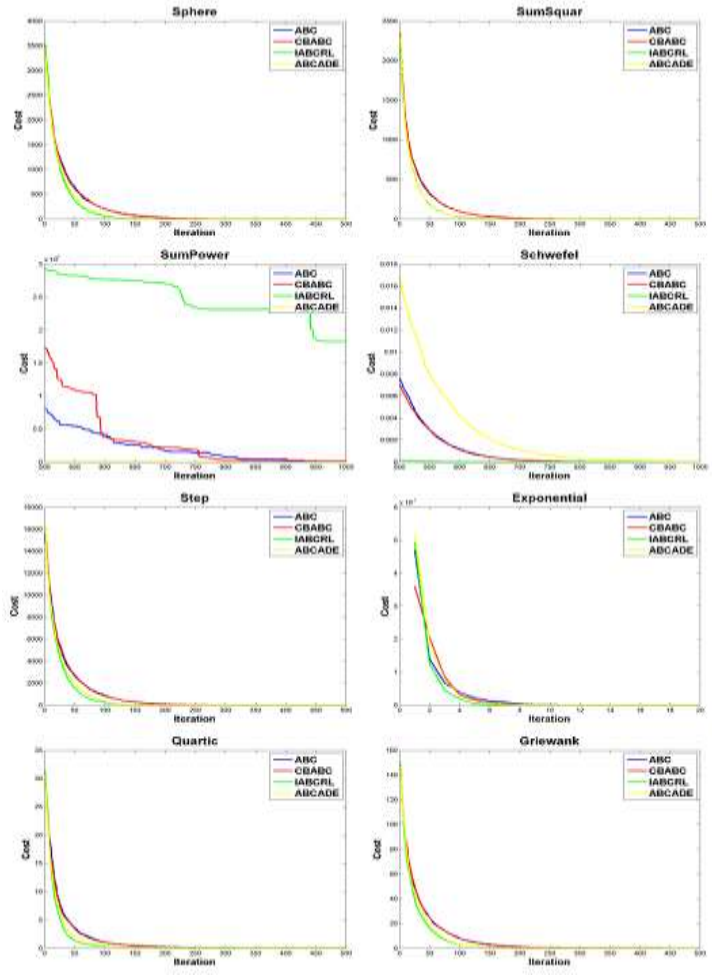
شکل ۲. روند همگرایی الگوریتم پیشنهادی و الگوریتم‌های مشابه برای آزمایش اول

جدول ۲. میانگین و انحراف معیار جواب یافت شده برای توابع معیار در آزمایش اول

نام تابع		ABC	ABC_ADE	CB_ABC	IABCRL
Sphere	میانگین	327.2735	313.9177	329.8998	**284.2001**
	انحراف معیار	6.1084e+06	5.9231e+06	5.8758e+06	**5.7194e+06**
SumSquare	میانگین	28.1260	27.5884	29.1343	**25.6329**
	انحراف معیار	**4.7885e+04**	4.8080e+04	4.9959e+04	5.0419e+04
SumPower	میانگین	1.0627e+12	1.5056e+12	**6.3777e+11**	1.1954e+12
	انحراف معیار	1.7092e+27	2.6232e+27	**5.8052e+26**	1.2963e+27
Schwefel 2.22	میانگین	3.6135e+04	**3.0073e+04**	3.8541e+04	2.1488e+05
Step	میانگین	326.2394	**251.4268**	329.7636	290.5760
	انحراف معیار	5.7689e+06	**4.7192e+06**	6.2066e+06	6.0879e+06
Exponential	میانگین	5.2282e-05	5.1098e-05	4.4356e-05	**4.4215e-05**
	انحراف معیار	1.3208e-06	1.7464e-06	**1.0311e-06**	1.2783e-06
Quartic	میانگین	0.3803	0.3630	0.3680	**0.2924**
	انحراف معیار	7.1208	7.2311	6.8643	**6.3314**
Griewank	میانگین	3.3368	3.0275	3.2903	**2.7302**
	انحراف معیار	533.2800	493.6781	501.2741	**488.7361**

۴-۴. آزمایش دوم

در این آزمایش برای تمامی توابع معیار جدول ۱، متغیر D=30 در نظر گرفته شده است. همچنین برای هر روش تعداد جمعیت جواب‌ها ۱۰ و تعداد گام‌ها ۲۰۰۰ در نظر گرفته شده است. در جدول ۳ میانگین و انحراف معیار جواب‌های یافت شده آزمایش دوم بعد از همگرایی برای الگوریتم پیشنهادی و الگوریتم‌های مشابه مذکور نشان داده شده است. همانطور که در جدول ۳ قابل مشاهده است، روش پیشنهادی در بهینه‌سازی توابع SumSquare، Step، Quartic، Griewank هم از نظر میانگین و هم از نظر انحراف معیار بهتر از سایر روش‌ها عمل کرده است. در بهینه‌سازی تابع Exponential نیز روش پیشنهادی از نظر میانگین بهتر از سایر روش‌ها عمل کرده است. در شکل ۳ نیز روند همگرایی روش پیشنهادی و سایر روش‌های مشابه نشان داده شده است. همانطور که در شکل ۳ قابل مشاهده است، روش پیشنهادی در ۵۰۰ تکرار اول در بهینه‌سازی توابع Sphere، Schwefel، Step، Exponential، Quartic و Griewank زودتر از سایر روش‌ها به جواب بهینه همگرا شده است.



شکل ۳. روند همگرایی الگوریتم پیشنهادی و الگوریتم‌های مشابه در آزمایش دوم

جدول ۳. میانگین و انحراف معیار جواب یافت شده برای توابع معیار در آزمایش دوم

1.0449e+03	2.0947e+03	2.2322e+03	2.0781e+03	میانگین	**Sphere**
1.2115e+07	1.8416e+07	1.8770e+07	1.7996e+07	انحراف معیار	
236.4858	407.5715	337.2812	398.1391	میانگین	**SumSquare**
6.7845e+05	8.7169e+05	8.3039e+05	8.4481e+05	انحراف معیار	
2.6597e+39	**1.2571e+38**	1.3042e+39	5.1854e+38	میانگین	**SumPower**
5.9605e+81	**7.2157e+78**	9.9917e+80	1.2852e+80	انحراف معیار	
5.5284e+20	**2.9906e+19**	2.0520e+20	7.3171e+19	میانگین	**Schwefel 2.22**
2.9768e+44	**6.7311e+41**	3.8217e+43	4.4379e+42	انحراف معیار	
1.1135e+03	2.1092e+03	2.2626e+03	2.1198e+03	میانگین	**Step**
1.2384e+07	1.8155e+07	1.9038e+07	1.8533e+07	انحراف معیار	
2.4210e-06	3.0627e-06	4.3614e-06	4.9095e-06	میانگین	**Exponential**
3.7082e-09	6.1248e-09	9.6137e-09	**1.3758e-08**	انحراف معیار	
2.5274	6.7129	4.6863	6.7081	میانگین	**Quartic**
93.7447	181.3488	181.3562	189.0799	انحراف معیار	
10.0598	18.8741	15.0617	18.9745	میانگین	**Griewank**
1.0537e+03	1.4203e+03	1.4425e+03	1.4422e+03	انحراف معیار	

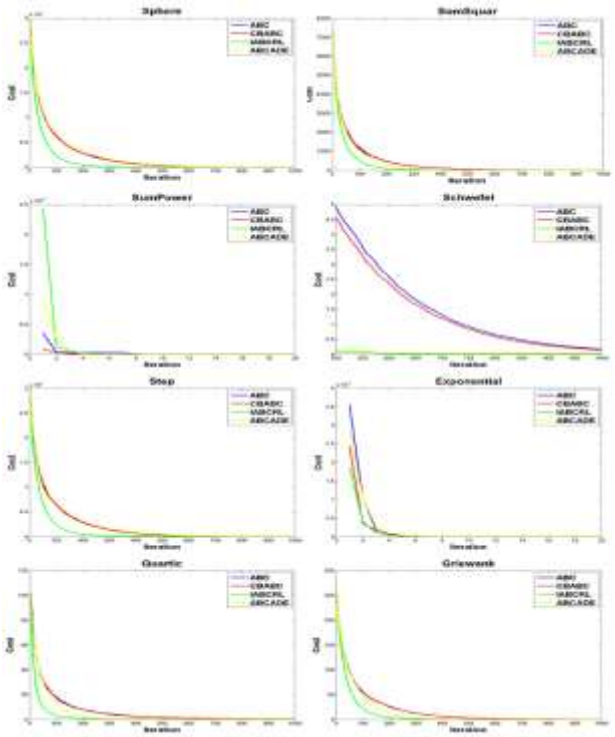
۴-۵. آزمایش سوم

در این آزمایش برای تمامی توابع معیار جدول ۱، متغیرD=50 در نظر گرفته شده است. همچنین برای هر روش تعداد جمعیت جواب‌ها ۱۰ و تعداد گام‌ها ۲۰۰۰ در نظر گرفته شده است. در جدول ۴ میانگین و انحراف معیار جواب های یافت شده آزمایش سوم بعد از همگرایی برای الگوریتم پیشنهادی و الگوریتم‌های مشابه مذکور نشان داده شده است. همانطور که در جدول ۴ قابل مشاهده است، روش پیشنهادی در بهینه‌سازی توابع Sphere، SumSquare، Step، Quartic، Griewank هم از نظر میانگین و هم از نظر انحراف معیار بهتر از سایر روش‌ها عمل کرده است. همچنین در بهینه‌سازی تابع Exponential، روش پیشنهادی از نظر میانگین بهتر از سایر روش‌ها عمل کرده است. در شکل ۴ نیز روند همگرایی روش پیشنهادی و سایر روش‌های مشابه نشان داده شده است. همانطور که در شکل ۴ قابل مشاهده است، روش پیشنهادی در ۱۰۰۰ تکرار اول در بهینه‌سازی توابع Sphere، SumSquar، Step، Quartic و Griewank زودتر از سایر روش‌ها به جواب بهینه همگرا شده است.

جدول ۴. میانگین و انحراف معیار جواب یافت شده برای توابع معیار در آزمایش سوم

1.0449e+03	2.0947e+03	2.2322e+03	2.0781e+03	میانگین	**Sphere**
1.2115e+07	1.8416e+07	1.8770e+07	1.7996e+07	انحراف معیار	
236.4858	407.5715	337.2812	398.1391	میانگین	**SumSquare**
6.7845e+05	8.7169e+05	8.3039e+05	8.4481e+05	انحراف معیار	
2.6597e+39	1.2571e+38	1.3042e+39	5.1854e+38	میانگین	**SumPower**
5.9605e+81	7.2157e+78	9.9917e+80	1.2852e+80	انحراف معیار	
5.5284e+20	2.9906e+19	2.0520e+20	7.3171e+19	میانگین	**Schwefel 2.22**
2.9768e+44	6.7311e+41	3.8217e+43	4.4379e+42	انحراف معیار	
1.1135e+03	2.1092e+03	2.2626e+03	2.1198e+03	میانگین	**Step**
1.2384e+07	1.8155e+07	1.9038e+07	1.8533e+07	انحراف معیار	
2.4210e-06	3.0627e-06	4.3614e-06	4.9095e-06	میانگین	**Exponential**
3.7082e-09	6.1248e-09	9.6137e-09	1.3758e-08	انحراف معیار	
2.5274	6.7129	4.6863	6.7081	میانگین	**Quartic**
93.7447	181.3488	181.3562	189.0799	انحراف معیار	
10.0598	18.8741	15.0617	18.9745	میانگین	**Griewank**
1.0537e+03	1.4203e+03	1.4425e+03	1.4422e+03	انحراف معیار	

همانطور که از مقایسه آزمایش‌ها فوق می‌توان نتیجه گرفت، روش پیشنهادی در بهینه‌سازی توابع هدف پیچیده که تعداد پارامتر مجهول بیشتری دارند زودتر از سایر روشها به جواب بهینه همگرا می‌شود.



شکل ۴. روند همگرایی الگوریتم پیشنهادی و الگوریتم‌های مشابه در آزمایش سوم

همان‌طور که ملاحظه می‌شود، با استفاده از یادگیری تقویتی تعادل مناسب بین اکتشاف و استخراج برقرار شده و لذا الگوریتم جواب‌های با کیفیت‌تری را تولید خواهد کرد. به نحوی که در اغلب موارد جواب‌ها از صحت بیشتری برخوردار بوده و تکرارپذیری بیشتری دارند یعنی انحراف معیار پاسخ‌ها کمتر شده است.

۵. نتیجه گیری و کارهای آینده

در این مقاله به حل مشکل الگوریتم کلونی زنبور عسل مصنوعی (ABC) مبنی بر به عدم قدرت کافی در استخراج جواب‌های پیرامون جواب‌های قبلی پرداخته شد. در روش ارائه شده، استفاده از الگوریتم یادگیری تقویتی (RL) در تعیین عملیات مناسب در هر موقعیت استفاده شده است. به عبارت دیگر در روش ارائه شده، از تعدادی عبارات بروز رسانی اکتشافی و استخراجی به صورت ترکیبی استفاده می‌شود بطوری که RL با توجه به شرایطی که ذره در آن قرار دارد، عبارت بروزرسانی مناسب را تعیین می‌کند. جهت بررسی عملکرد روش پیشنهادی و مقایسه آن با سایر الگوریتم‌های تکاملی، از آن‌ها در توابع معیار متعددی استفاده شده است که نتیجه آزمایش‌ها نشان دهنده قدرت روش پیشنهادی در مقایسه با سایر الگوریتم‌های تکاملی است. اما از آنجا که در روش پیشنهادی، یک فاز جدید جهت تعیین عملیات بروزرسانی وجود دارد، در این روش نسبت به روش‌های مشابه شاهد افزایش زمان اجرا هستیم.

جهت بهبود الگوریتم کارایی الگوریتم پیشنهادی از دو جنبه نیاز به بررسی دقیق تر هستیم. جنبه اول، تعیین دقیق عبارات بروز رسانی بطوری که به وسیله آن‌ها بتوانیم به طور همزمان ویژگی‌های اکتشاف، استخراج و همگرایی به جواب بهینه را تعیین کنیم. جنبه دوم کاهش زمان اجرای روش پیشنهادی است. برای اینکار می‌توان به جای اینکه هر یک از ذره‌ها از یک ماتریس یادگیری جداگانه‌ای استفاده کنند، از چند ماتریس یادگیری برای تمامی ذره‌ها استفاده شود. با این کار می‌توان حجم محاسبات تعیین استراتژی مناسب در هر لحظه را کاهش داد.

مراجع

[1] P. Rakshit *et al.*, "Realization of an adaptive memetic algorithm using differential evolution and q-learning: a case study in multirobot path planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 814-831, 2013.

[2] G. Chen, J. J. C. i. Liu, and neuroscience, "Mobile Robot Path Planning Using Ant Colony Algorithm and Improved Potential Field Method," vol. 2019, 2019.

[3] K. Pandremmenou, L. P. Kondi, and K. E. Parsopoulos, "A study on visual sensor network cross-layer resource allocation using quality-based criteria and metaheuristic optimization algorithms," *Applied Soft Computing*, vol. 26, pp. 149-165, 2015.

[4] S. C. Chiam, K. C. Tan, and A. A. Mamun, "A memetic model of evolutionary PSO for computational finance applications," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3695-3711, 2009.

[5] M. Rahmani, M. K. Eraqi, H. J. A. i. M. F. Nikoomaram, and Applications, "Portfolio Optimization by Means of Meta Heuristic Algorithms," vol. 4, no. 4, pp. 83-97.2019 ,

[6] S.-H. Yang and J.-F. Kiang, "Optimization of asymmetrical difference pattern with memetic algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 4, pp. 2297-2302, 2014.

[7] M. Sornam and M. Prabhakaran, "Logit-Based Artificial Bee Colony Optimization (LB-ABC) Approach for Dental Caries Classification Using a Back Propagation Neural Network," in *Integrated Intelligent Computing, Communication and Security*: Springer, 2019, pp. 79-91.

[8] J. Gao, Y. Lu, J. Qi, and L. J. I. A. Shen" ,A Radar Signal Recognition System Based on Non-Negative Matrix Factorization Network and Improved Artificial Bee Colony Algorithm," vol. 7, pp. 117612-117626, 2019.

[9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department2005.

[10] Z. Ding, M. Huang, and Z. Lu, "Structural damage detection using artificial bee colony algorithm with hybrid search strategy," *Swarm and Evolutionary Computation*, vol. 28, pp. 1-13, 2016.

[11] Y. Cao, Y. Lu, X. Pan, and N. J. C. c. Sun, "An improved global best guided artificial bee colony algorithm for continuous optimization problems," vol. 22, no. 2, pp. 3011-3019, 2019.

[12] D. Karaboga and B. Gorkemli" ,A quick artificial bee colony (qABC) algorithm and its performance on optimization problems," *Applied Soft Computing*, vol. 23, pp. 227-238, 2014.

[13] I. Brajevic, "Crossover-based artificial bee colony algorithm for constrained optimization problems," *Neural Computing and Applications*, vol. 26, no. 7, pp. 1587-1601, 2015.

[14] Z. Zhu, J. Zhou, Z. Ji, and Y.-H. Shi, "DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm," *IEEE Transactions on Evolutionary Computation* ,vol. 15, no. 5, pp. 643-658, 2011.

[15] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871-882, 2011.

[16] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5682-5687, 2010.
[17] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120-142, 2012
[18] Cui L, Li G, Wang X, Lin Q, Chen J, Lu N, Lu J (2017) "A ranking-based adaptive artificial bee colony algorithm for global numerical optimization." Inf Sci 417:169–185
[19] A. Abraham, R. K. Jatoth, and A. Rajasekhar, "Hybrid differential artificial bee colony algorithm," *Journal of computational and theoretical Nanoscience*, vol. 9, no. 2, pp. 249-257, 2012.
[20] W. Xiang, S. Ma, and M. An, "Habcde: a hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution," *Applied Mathematics and Computation*, vol. 238, pp. 370-386, 2014.
[21] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive ", *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945-958, 2009.
[22] Z. Liang, K. Hu, Q. Zhu, and Z. Zhu, "An enhanced artificial bee colony algorithm with adaptive differential operators," *Applied Soft Computing*, vol. 58, pp. 480-494, 2.017
[23] X. Song, Q. Yan, and M. Zhao, "An adaptive artificial bee colony algorithm based on objective function value information," *Applied Soft Computing*, vol. 55, pp. 384-401, 2017
[24] X Song, M Zhao, Q Yan, S Xing - Swarm and Evolutionary Computation, 2019 - Elsevier " A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization"
[25] G. S. Piperagkas, G. Georgoulas, K. E. Parsopoulos, C. D. Stylios ,and A. C. Likas, "Integrating particle swarm optimization with reinforcement learning in noisy problems," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, 2012, pp. 65-72: ACM.
[26] H. Samma, C. P. Lim, and J. M. Saleh, "A new reinforcement learning-based memetic particle swarm optimizer," *Applied Soft Computing*, vol. 43, pp. 276-297, 2016.
[27] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied mathematics and computation*, vol. 217, no. 7, pp. 3166-3173, 2010.
[28] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences*, vol. 300, pp. 140-157, 2015.
[29] F. Marcelloni and M. Vecchio, "Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization," *Information Sciences*, vol. 180, no. 10, pp. 1924-1941, 2010.
180, no. 10, , 2010. pp. 1924-1941
[30] Gao H, Fu Z, Pun CM, Hu H, Lan R. A multi-level thresholding image segmentation based on an improved artificial bee colony algorithm. Computers & Electrical Engineering. 2018 Aug 1;70:931-8.
[31] Zhao H, Zhang C. A decomposition-based many-objective artificial bee colony algorithm with reinforcement learning. Applied Soft Computing. 2020 Jan 1;86:105879.
[32] Romero-Hdz J, Saha BN, Tstutsumi S, Fincato R. Incorporating domain knowledge into reinforcement learning to expedite welding sequence optimization. Engineering Applications of Artificial Intelligence. 2020 May 1;91:103612.
[33] Fairee S, Prom-On S, Sirinaovakul B. Reinforcement learning for solution updating in Artificial Bee Colony. PloS one. 2018 Jul 17;13(7):e0200738.
[34] Yu J, You X, Liu S. Dynamic reproductive ant colony algorithm based on piecewise clustering. Applied Intelligence. 2021 Apr 12:1-21.
[35] Lu R, Hu H, Xi M, Gao H, Pun CM. An improved artificial bee colony algorithm with fast strategy, and its application. Computers & Electrical Engineering. 2019 Sep 1;78:79