

الگوریتم بهینه سازی فاخته

رامین رجبیون

قطب کنترل و پردازش هوشمند، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران

چکیده

ماندن، به حالتی برسد که تنها یک جامعه از فاخته ها وجود داشته باشد، همه با یک ارزش سود مشابه. استفاده از الگوریتم

پیشنهاد شده در چند تابع محک و یک مسئله واقعی، توانایی آن را در حل مسائل دشوار بهینه سازی ثابت می کند.

1. معرفی

بهینه سازی روند بهتر کردن چیزی است. به عبارت دیگر، بهینه سازی روند تنظیم ورودی ها، مشخصه های یک وسیله، فرآیند ریاضیاتی، یا آزمایش برای یافتن کمینه یا بیشینه خروجی یا نتیجه است. ورودی از یک سری متغیرها تشکیل شده است: روند یا تابع که به عنوان تابع هزینه شناخته می شود، تابع هدف یا تابع برازندگی؛ و خروجی نیز هزینه یا برازندگی [1] است. روش های متفاوتی برای حل یک مسئله بهینه سازی وجود دارد. برخی از این روش ها از رویه های طبیعی الهام گرفته اند. این روش ها معمولاً

در این مقاله یک الگوریتم تکاملی جدید، مناسب برای مسائل بهینه سازی غیرخطی پیوسته، معرفی می شود. این الگوریتم بهینه سازی از زندگی یک خانواده از پرندگان به نام فاخته ها، الهام گرفته است. زندگی خاص این نوع پرندگان و ویژگی های آن در تخم گذاری و زاد و ولد، انگیزه اصلی پرورش این الگوریتم تکاملی بهینه سازی جدید بوده است. مانند سایر روش های تکاملی، الگوریتم بهینه سازی فاخته (COA)، با یک جمعیت اولیه آغاز می شود. جمعیت فاخته ها، در جوامع مختلف آن ها، دو نوع است: فاخته های بالغ و تخم ها. تلاش فاخته ها برای زنده ماندن اساس الگوریتم بهینه سازی فاخته را تشکیل می دهد. در طول مسابقه برای زنده ماندن، برخی از فاخته ها یا تخم های آنان از بین می روند. جامعه فاختگان نجات یافته به یک محیط بهتر مهاجرت می کنند، و دوباره شروع به تکثیر و تخم گذاری می کنند. امید است تلاش فاخته ها برای زنده

¹ Cuckoo Optimization Algorithm

با مجموعه اولیه ای از متغیرها آغاز می شوند و سپس جلو می روند تا زمانی که حداقل یا حداکثر مطلق تابع هدف به دست آید. الگوریتم ژنتیک¹ (GA) محبوب ترین تکنیک در زمینه پژوهش های محاسبات تکاملی بوده است. الگوریتم ژنتیک از عواملی بهره می برد که توسط تنوع ژنتیک طبیعی و گزینش طبیعی الهام بخشیده شده است [2,3]. مثال دیگر، بهینه سازی ازدحام ذرات² (PSO) است که در سال 1995 توسط ابرهارت و کندی ایجاد شد. این الگوریتم تصادفی از رفتار جمعی دسته پرندگان، یا حرکت جمعی ماهی ها، الهام گرفته است [3-5]. بهینه سازی کلونی مورچگان³ (ACO) یک الگوریتم تکاملی بهینه سازی دیگر است که از برجا گذاشتن اثر فرومون در کلونی های واقعی مورچه ها، الهام گرفته است [3,6,7]. از طرف دیگر، آنیلینگ شبیه سازی شده، فرآیند آنیلینگ را که در آن یک ماده تا بالای نقطه ذوبش گرم می شود و سپس به آهستگی سرد می شود تا یک شبکه کریستالی تشکیل شود؛ شبیه سازی می کند، و توزیع احتمال انرژی اش را حداقل می کند [1,8,9]. در کنار این روش های شناخته شده، بررسی ها روی الگوریتم های بهینه سازی الهام گرفته از طبیعت، همچنان در حال انجام هستند و روش های جدیدی در حال گسترش اند تا نوعی از مسائل غیر خطی را حل کنند. در [10]، استفاده از ارگودیسیتیه و تصادفی بودن درونی تکرر بی نظمی، یک الگوریتم تکاملی ایمن جدید

بر اساس الگوریتم بهینه سازی بی نظمی و الگوریتم تکاملی ایمن معرفی شده تا همگرایی الگوریتم تکاملی ایمن را بهبود بخشد. این الگوریتم جدید مزایای الگوریتم تکاملی ایمن و الگوریتم بهینه سازی بی نظمی را در هم می آمیزد. [11] یک تکنیک بهینه سازی جدید را معرفی می کند که روش انفجار نارنجک⁴ (GEM) نامیده می شود و نظرات اساسی آن، شامل مفهوم جهت جست و جوی بهینه⁵ (OSD) با جزئیات شرح داده شده اند. در [12] یک روش جدید بهینه سازی ازدحام ذرات بر اساس الگوریتم انتخاب کلونال، پیشنهاد شده تا از همگرایی زودهنگام پرهیز، و گوناگونی جمعیت را تضمین کند.

مزایای اصلی الگوریتم های تکاملی شامل موارد زیر است [3]:

(1) مقاوم بودن در برابر تغییرات پویا: روش های سنتی بهینه سازی در برابر تغییرات پویا در محیط مقاوم نیستند و برای یافتن یک راه حل نیاز به یک شروع مجدد کامل دارند. به عکس، از محاسبات تکاملی می توان برای سازگار کردن راه حل ها با شرایط متغیر استفاده کرد.

(2) کاربرد وسیع: الگوریتم های تکاملی قابلیت آن را دارند که برای هر مسئله ای که بتوان آن را با مسائل بهینه سازی تابع فرموله کرد، به کار برده شوند.

(3) پیوند یافتن با سایر روش ها: الگوریتم های تکاملی می توانند با تکنیک های بهینه سازی قدیمی ترکیب شوند.

⁴ Grenade Explosion Method

⁵ Optimal Search Direction

¹ Genetic Algorithm

² Particle Swarm Optimization

³ Ant Colony Optimization

(4) حل مسائلی که هیچ راه حلی ندارند: یک مزیت الگوریتم های تکاملی توان درگیر شدن با مسائلی است که برای آن ها هیچ مهارت انسانی وجود ندارد. اگرچه مهارت انسانی زمانی که مورد نیاز و در دسترس است، باید به کار گرفته شود، اغلب برای شیوه های حل مسئله خودکار، ناکافی است.

با توجه به این مشخصات، می توان الگوریتم های تکاملی را در مسائل مختلفی اعمال کرد؛ شامل عملیات و کنترل سیستم های قدرت [13، 19، 20]، مسائل ترکیباتی ان پی- سخت [14، 15]، فرآیند های شیمیایی [16]، مسائل برنامه ریزی شغلی [17]، مسائل مسیریابی و سایل نقلیه، شبکه بندی موبایل، برنامه ریزی پردازش دسته ای، مسائل بهینه سازی چند هدفه [18]، مدل سازی پارامترهای بهینه [21]، پردازش تصویر و مسائل بازشناخت الگو.

در این مقاله، ما یک الگوریتم تکاملی بهینه سازی جدید را معرفی می کنیم که از زندگی یک خانواده از پرندگان به نام "فاخته" الهام گرفته است. نحوه خاص تخم گذاری و زاد و ولد فاخته ها، اساس این الگوریتم بهینه سازی نوین است. فاخته هایی که در این مدل از آن ها استفاده می شود، بر دو نوع اند: فاخته های بالغ و تخم ها. فاخته های بالغ در لانه های پرندگان دیگر تخم گذاری می کنند و اگر این تخم ها توسط پرنده های میزبان شناسایی و نابود نشوند، رشد می کنند و به فاخته های بالغ تبدیل می شوند. امید می رود، خصوصیات زیست محیطی و مهاجرت جوامع (گروه های) فاخته ها آن ها را به سمت همگرایی و یافتن بهترین

محیط برای زاد و ولد و تکثیر هدایت کند. این بهترین محیط، ماکزیمم مطلق توابع هدف است. این مقاله نشان می دهد که چگونه شیوه حیات فاخته ها مدل سازی و اجرا می شود.

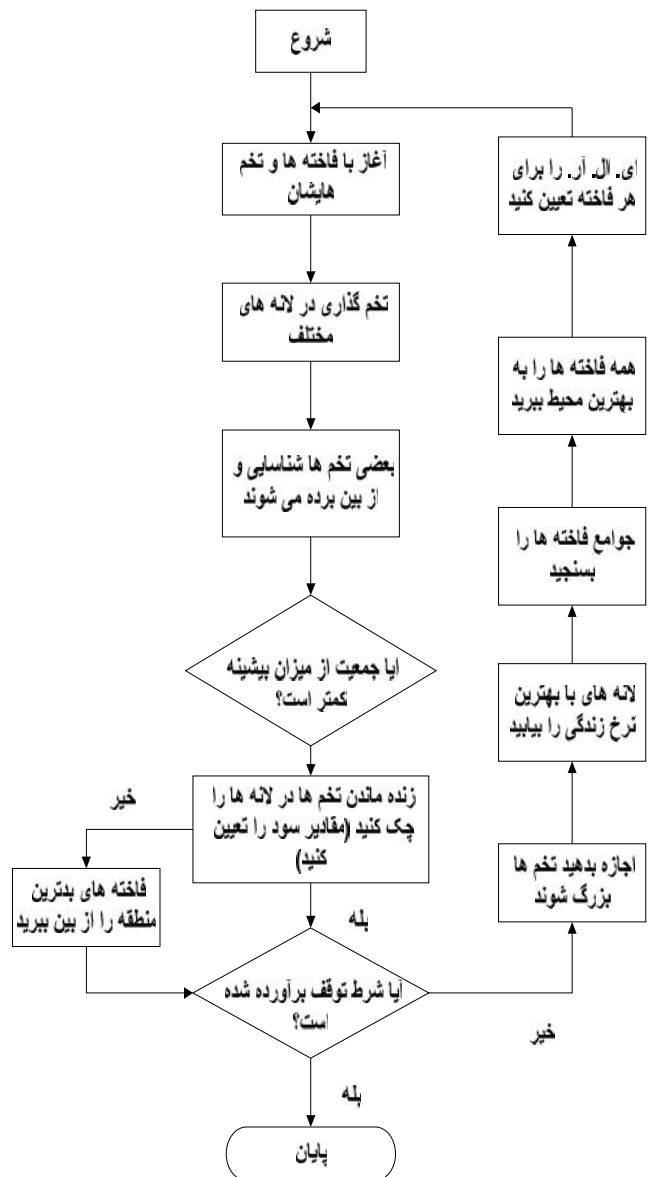
بخش 2 پرندگان به نام فاخته را مورد بررسی قرار می دهد و ویژگی های شگفت انگیز زندگی آن ها را مرور می کند. در بخش 3، الگوریتم بهینه سازی فاخته ارائه می شود و قسمت های مختلف آن با جزئیات مورد مطالعه قرار می گیرد. در بخش 4 الگوریتم پیشنهاد شده با چند تابع محک و نیز با طراحی کنترل کننده یک فرآیند چند ورودی - چند خروجی (MIMO) به عنوان یک مطالعه موردی واقعی، آزموده می شود. سرانجام در بخش 5 نتیجه گیری انجام می شود.

2. فاخته ها و روش منحصر به فرد آن ها در تکثیر

همه 9000 گونه پرندگان شیوه یکسانی برای والد شدن دارند: روش تخم گذاری. هیچ پرنده ای نوزاد به دنیا نمی آورد. پرندگان به سرعت یک تخم درون یک پوسته محافظ تشکیل می دهند و روی آن می خوابند. اندازه بزرگ تخم ها، این را برای پرنده مؤنث دشوار می کند که در یک زمان، از بیش از یک تخم نگهداری کند. به همراه داشتن تخم ها، پرواز را سخت تر می کند و به انرژی بیشتری نیاز دارد. و چون یک تخم، غنیمتی پر از پروتئین و مغذی برای همه شکارچیان است، پرندگان باید جایی امن برای تخم گذاری بیابند. یافتن مکانی برای تخم -

می سازند تا از شناسایی توسط شکارچیان دور بمانند. بعضی از آنان چنان در پنهان کردن لانه هایشان موفق اند که حتی چشم های جست و جوگر انسان هم به ندرت آن ها را دیده است.

پرندگان هم هستند که تمام عرف های لانه سازی و والدی را رها می کنند و برای پروراندن خانواده شان به حيله گری متوسل می شوند. این ها "جوجه های انگلی" هستند، پرندگانی که هرگز لانه خودشان را نمی سازند و در عوض در لانه گونه های دیگر تخم گذاری می کنند و نگه داری از جوجه شان را به والدین آن گونه می سپارند. فاخته، شناخته شده ترین انگل جوجه است و ماهر در فریب ظالمانه. رویکرد او شامل پنهان کاری، غافل گیری و سرعت است. فاخته مادر تخم پرند گونه میزبان را برمی دارد، تخم خودش را به جای آن در لانه قرار می دهد و با تخم میزبان روی منقارش لانه را ترک می کند. کل این فرآیند کم تر از 10 ثانیه طول می کشد. فاخته ها انگل لانه های انواع زیادی از گونه های پرندگان می شوند و با دقت رنگ و شکل تخم های خود را با تخم های گونه های میزبان تطبیق می دهند. هر فاخته مؤنث روی یک گونه میزبان خاص، تخصص دارد. این که چگونه فاخته قادر است طوری تخم گذاری کند که دقیقاً از تخم های هر میزبان تقلید کند، یکی از معماهای اصلی طبیعت است. بسیاری از پرندگان یاد می گیرند که یک تخم فاخته قرار گرفته در لانه خودشان را شناسایی کنند و یا این تخم عجیب را به بیرون پرتاب کنند یا آن لانه را رها کنند و از



شکل 1. فلوچارت الگوریتم بهینه سازی فاخته

گذاری به صورت ایمن و پرورش جوجه شان تا رسیدن به استقلال، چالشی است که پرندگان آن را با راه های هوشمندانه ای حل کرده اند. آن ها از هنر، طراحی و مهندسی پیچیده استفاده می کنند. تنوع ساختاری لانه ها هیچ معادلی در دنیای جانوران ندارد. بسیاری از پرندگان، لانه های منفرد و ناپیدا، پنهان شده لابلای پوشش گیاهی

نولانه بسازند. بنابراین فاخته دائما تلاش دارد در تقلید از تخم های میزبان، پیشرفت کند در حالی که پرندگان میزبان سعی در یافتن راه هایی برای شناسایی تخم انگل دارند. تقلای میان میزبان و انگل مشابه یک مسابقه تسلیحاتی است که در آن هر طرف سعی در غلبه بر دیگری در زنده ماندن دارد [22].

برای فاخته ها محل زندگی مناسب جایی است که منبعی از غذا - به طور کلی حشرات و مخصوصا کرم ابریشم - و جایی برای تخم گذاری را فراهم کند؛ برای جوجه های انگلی مکان مورد نیاز، همان محل زندگی مناسب گونه های میزبان است. فاخته ها در مکان های بسیار گوناگونی زندگی می کنند. اکثر این پرنده ها در جنگل ها و درختستان ها وجود دارند، بیشتر در جنگل های همیشه سبز استوا. علاوه بر جنگل ها بعضی از گونه های فاخته در فضاهای بازتر زندگی می کنند؛ این فضاها حتی می تواند شامل مناطق خشک، مثل کویر باشد. گونه های مهاجر به مناطق معتدل همچون فاخته معمولی در طیف وسیعی از زیست بوم ها سکونت دارند تا بتوانند حداکثر استفاده از میزبان های بالقوه جوجه را داشته باشند؛ از نزارها گرفته تا صحراهای خالی از درخت.

بیشتر گونه های فاخته غیر مهاجر هستند، ولی انواع بسیاری از این پرندگان مهاجرت های فصلی منظم دارند، و بسیاری دیگر مهاجرت های جزئی در قسمتی از محدوده خودشان دارند. این مهاجرت ها ممکن است روزانه باشد، مثل فاخته منقار بلند، یا شبانه باشد، مثل فاخته نوک زرد. برای گونه

هایی که در عرض جغرافیایی بالاتر تخم گذاری می کنند، دسترسی به غذا آن ها را به مهاجرت به نواحی گرم تر در طول زمستان مجبور می کند، و همه این کار را می کنند. فاخته دم دراز که در نیوزیلند زاد و ولد می کند، به قشلاق هایش در پلی نزی، میکرونزی و ملانزی می رود؛ شاهکاری که شاید بتوان آن را چشمگیر ترین مهاجرت روی آب میان پرندگان خشکی زی توصیف کرد [23]. فاخته نوک زرد و فاخته نوک سیاه در آمریکای شمالی زاد و ولد می کنند و از این سو به آن سوی دریای کارائیب پرواز می کنند، یک پرواز بدون توقف 4000 کیلومتری. مهاجرت های طولانی دیگر، شامل فاخته صغیر است که از هند تا کنیا، روی اقیانوس هند (به اندازه 3000 کیلومتر) پرواز می کند؛ و فاخته های معمولی اروپایی که در سفرشان به آفریقای جنوبی، روی دریای مدیترانه و صحرای بزرگ آفریقا بدون توقف پرواز می کنند. درون آفریقا 10 گونه از فاخته ها مهاجرت های درون قاره ای منظمی دارند که قطبی نامیده می شود؛ به این صورت که آن ها فصل بدون زاد و ولد را در مرکز استوایی این قاره می گذارند و به منظور زاد و ولد در زمین های خشک تر و بازتر و صحراها، به شمال و جنوب می روند [24].

نزدیک 56 گونه از فاخته بر قدیم و 3 گونه از فاخته بر جدید جوجه های انگلی هستند، که تخم های شان را در لانه های سایر پرندگان قرار می دهند [25]. این گونه ها جوجه های انگلی ضروری هستند، به این معنی که تنها به این شیوه می توانند تولید مثل کنند. جوجه فاخته زودتر از جوجه میزبان

سر از تخم بیرون می آورد و زودتر رشد می کند. در بیشتر موارد، جوجه فاخته حتی تخم ها و جوجه های گونه میزبان را از لانه بیرون می اندازد. جوجه فرصتی برای آموختن این کار نداشته است، پس این باید یک گزینه منتقل شده با ژنتیک باشد. جوجه فاخته میزبان را با صدای تند درخواستش [26] و دهان بازش که یک علامت محرک است [27]؛ به همراه شدن با سرعت بالای رشدش تشویق می کند. انگل های فاخته مؤنث در قرار دادن تخم هایی که شباهت زیادی به تخم های میزبان منتخب شان دارند، تخصص دارند. این تخصص، یک گزینش طبیعی است؛ همان طور که بعضی پرندگان می توانند تخم های فاخته را از تخم های خودشان تمیز دهند؛ تخم هایی که کم ترین شباهت را به تخم هایی که از لانه بیرون انداخته شده اند، دارند. [27] گونه های میزبان ممکن است در دفعه اول فعالیت های مستقیم تری برای جلوگیری از تخم گذاری فاخته ها در لانه های شان انجام دهند. در نظر فاخته ها، پرندگانی که لانه شان بیشتر در معرض خطر اشغال توسط فاخته ها است، آن ها را از منطقه بیرون خواهند کرد [28]. فاخته های انگل به گروه هایی تقسیم می شوند که هر گروه در زمینه یک گونه میزبان خاص تخصص دارد. شواهدی وجود دارد مبنی بر این که این گروه ها از نظر ژنتیکی با یک دیگر متفاوت اند. تخصص در زمینه گونه های میزبان با نیاز به تقلید از تخم های آن ها افزایش می یابد.

3. الگوریتم پیشنهاد شده: الگوریتم بهینه سازی فاخته

شکل 1 فلوچارتی از الگوریتم پیشنهاد شده را نشان می دهد. مانند سایر الگوریتم های تکاملی، این الگوریتم نیز با یک جمعیت اولیه از فاخته ها شروع می شود. این فاخته های اولیه تخم هایی برای قرار دادن در لانه های پرندگان میزبان دارند. برخی از این تخم ها که شباهت بیشتری به تخم های پرنده میزبان دارند، فرصت رشد کردن و تبدیل به فاخته بالغ شدن را دارند. تخم های دیگر توسط پرنده های میزبان شناسایی و از بین برده می شوند. تخم های رشد یافته مناسب بودن لانه های آن منطقه را نشان می دهند. در یک منطقه، هر چه تخم های بیشتری زنده بمانند، سود بیشتری در آن منطقه به دست می آید. پس مکانی که تخم های بیشتری در آن زنده بمانند چیزی که است که COA در پی بهینه کردن آن است.

فاخته ها به دنبال مناسب ترین منطقه برای تخم گذاری هستند تا نرخ زنده ماندن تخم های شان را بیشینه سازند. پس از آن که تخم های باقی مانده رشد کردند و به فاخته بالغ بدل شدند، جوامعی را تشکیل می دهند. هر جامعه زیست بوم خودش را دارد. بهترین محل زندگی یافت شده فاختگان، مقصد فاخته ها در جوامع دیگر خواهد بود. سپس آن ها به این بهترین محل زندگی مهاجرت می کنند. آن ها در جایی نزدیک بهترین زیست بوم ساکن می شوند. با توجه به تعداد تخم های هر فاخته، و همچنین فاصله فاخته از نقطه مقصد (بهترین زیست بوم) یک شعاع تخم گذاری به آن داده می شود. سپس، فاخته شروع به خواباندن تخم هایش در لانه های تصادفی درون شعاع تخم گذاری اش

زندگی، ماتریس $1 \times N_{var}$ است که موقعیت فعلی فاخته را نشان می دهد. این ماتریس به شکل زیر تعریف می شود:

$$\text{habitat} = [X_1, X_2, \dots, X_{N_{var}}]$$

هر کدام از مقادیر متغیرها $(X_1, X_2, \dots, X_{N_{var}})$ یک عدد ممیز شناور هستند. سود یک محل زندگی توسط تابع سود f_p در یک محل زندگی $(X_1, X_2, \dots, X_{N_{var}})$ محاسبه می شود. بنابراین:

$$\text{سود} = f_p(\text{habitat}) = f_p(X_1, X_2, \dots, X_{N_{var}})$$

همان طور که دیده می شود، COA الگوریتمی است که تابع سود را ماکزیمم می کند. برای استفاده از این الگوریتم در مسائل مینیمم سازی هزینه ها، می توان به سادگی تابع سود زیر را ماکزیمم کرد:

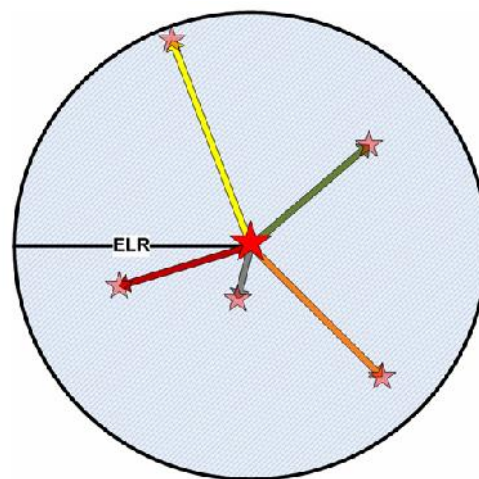
$$\text{سود} = -\text{Cost}(\text{habitat}) = -f_c(X_1, X_2, \dots, X_{N_{var}})$$

برای شروع الگوریتم بهینه سازی، ماتریس محل زندگی $N_{pop} \times N_{var}$ ایجاد می شود. سپس یک تعداد تصادفی از تخم ها برای هر کدام از این محل زندگی اولیه، پیشنهاد می شود. در طبیعت، هر فاخته بین 5 تا 20 تخم می گذارد. این مقادیر حدود بالا و پایین اختصاص تخم به هر فاخته در دفعات مختلف است. عادت دیگر فاخته های واقعی این است که در جایی با بیشترین فاصله از محل زندگی شان

می کند. این فرآیند ادامه می یابد تا بهترین موقعیت با بالاترین ارزش سود به دست آید و اکثر جمعیت فاخته ها در این مکان جمع شوند.

3.1. ایجاد محل زندگی اولیه فاخته

برای حل یک مسئله بهینه سازی، لازم است مقادیر متغیرهای مسئله به شکل یک ماتریس درآیند. در اصطلاحات مربوط به الگوریتم ژنتیک و روش بهینه سازی ازدحام ذرات، این ماتریس به ترتیب کروموزوم¹ و موقعیت ذره² نامیده می شود. اما این جا در الگوریتم



شکل 2. تخم گذاری تصادفی در ELR، ستاره قرمز در مرکز محل زندگی اولیه فاخته با 5 تخم است؛ ستاره های صورتی لانه های جدید برای تخم ها هستند.

بهینه سازی فاخته، این ماتریس محل زندگی³ نامیده می شود. در یک مسئله بهینه سازی N_{var} -بعدی، یک محل

³ habitat

¹ Chromosome

² Particle Position

تغذیه می کنند. نکته جالب توجه دیگر درباره تخم های فاخته، این است که تنها یک تخم در هر لانه فرصت رشد می یابد. این به خاطر آن است که وقتی تخم فاخته می شکند و جوجه فاخته سر از تخم بیرون می آورد، تخم های پرنده میزبان را از لانه بیرون می اندازد. در صورتی که تخم های پرنده میزبان زودتر از تخم فاخته بشکنند، جوجه فاخته بیشتر غذایی که پرنده میزبان به لانه می آورد را می خورد (چون جثه اش سه برابر بزرگ تر است، باقی جوجه ها را کنار می زند و بیشتر می خورد). بعد از چند روزی جوجه های خود پرنده میزبان می میرند و فقط جوجه فاخته در لانه می ماند.

3.3 مهاجرت فاخته ها

هنگامی که جوجه های فاخته رشد می کنند و بالغ می شوند، برای مدتی در منطقه و جامعه خودشان می مانند. اما وقتی به زمان تخم گذاری نزدیک می شوند، به محل زندگی جدید و بهتر مهاجرت می کنند.

در این محل ها، تخم های فاخته شباهت بیشتری به تخم پرندگان میزبان دارند و غذای بیشتری نیز برای جوجه های جدید وجود دارد. پس از شکل گرفتن گروه های فاخته در مناطق مختلف، جامعه با بهترین مقدار سود به عنوان نقطه

تخم گذاری می کنند. از این جا به بعد، این فاصله بیشینه شعاع تخم گذاری¹ (ELR) نامیده می شود. در یک مسئله بهینه سازی با حد بالای var_{hi} و حد پایین var_{low} برای متغیر ها، هر فاخته یک شعاع تخم گذاری دارد که با تعداد کل تخم ها، تعداد تخم های فعلی آن فاخته و نیز حدود var_{hi} و var_{low} متناسب است. بنابراین ELR به صورت زیر است:

$$ELR = \alpha \times \frac{\text{تعداد تخم های فاخته}}{\text{تعداد کل تخم ها}} \times (var_{hi} - var_{low})$$

که α عدد صحیح است که برای به دست آوردن مقدار ماکزیمم ELR به کار می رود.

3.2 روش تخم گذاری فاخته ها

هر فاخته با ELR خود، شروع به تخم گذاری در بعضی از لانه های پرنده های میزبان می کند. او این کار را به صورت تصادفی انجام می دهد. شکل 2 نمای واضحی از این مفهوم را نمایش می دهد.

پس از آن که همه تخم های فاخته ها در لانه های پرنده های میزبان قرار گرفتند، بعضی از آن ها که کمتر شبیه تخم های خود پرنده میزبان هستند، توسط آن ها شناسایی و بیرون انداخته می شوند. بنابراین بعد از فرآیند تخم گذاری، $p\%$ از تخم ها (معمولاً 10٪) با مقادیر سود کم تر، نابود می شوند. این تخم ها شناسی برای رشد کردن ندارند. مابقی تخم ها در لانه های میزبان بزرگ می شوند، جوجه ها سر از تخم بیرون می آورند و پرندگان میزبان آن ها را

¹ Egg Laying Radius

به اندازه \emptyset رادیان دارد. این دو پارامتر، λ و \emptyset به فاخته ها کمک می کند موقعیت های بسیار بیشتری را در محیط جست و جو کنند. برای هر فاخته، λ و \emptyset به شکل زیر هستند:

$$\lambda \sim U(0,1)$$

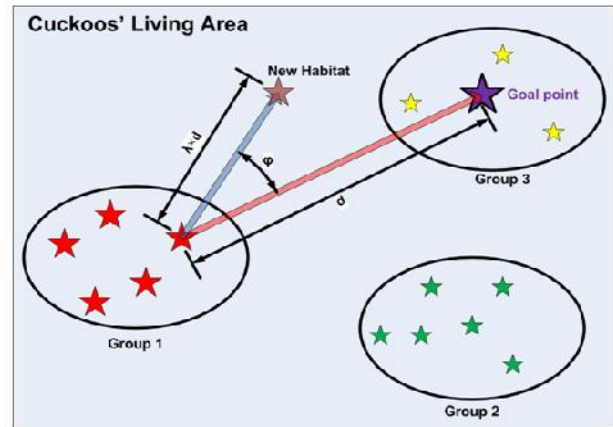
$$\emptyset \sim U(-\omega, \omega)$$

که $\lambda \sim U(0,1)$ یعنی λ یک عدد تصادفی با توزیع یکنواخت بین 0 و 1 است. ω پارامتری است که میزان انحراف از مقصد را محدود می کند. $\omega = \pi/6$ rad برای همگرایی مناسب جمعیت فاخته، برای سود کلی بیشینه، کافی به نظر می رسد.

وقتی همه فاخته ها به نقطه هدف مهاجرت کردند و محل زندگی جدید آنها مشخص گردید، به هر فاخته بالغ چند تخم داده می شود. سپس با توجه به تعداد تخم های اختصاص داده شده به هر پرنده، یک ELR برای هر فاخته محاسبه می شود. پس از آن فرآیند جدید تخم گذاری از نو آغاز می شود.

3.4. حذف فاخته ها در بدترین محل های زندگی

با توجه به این واقعیت که همیشه در جمعیت پرنده ها، تعادل برقرار است، بنابراین عدد N_{max} تعداد بیشینه فاخته های زنده در محیط را کنترل و محدود می کند. این تعادل به علت محدودیت های غذایی، کشته شدن توسط شکارچیان و همچنین عدم توانایی بر یافتن لانه مناسب برای تخم ها به وجود می آید. در مدل سازی پیشنهاد شده در



شکل 3. مهاجرت یک فاخته به محل زندگی هدف

هدف برای سایر فاخته ها برگزیده می شود تا به آن مهاجرت کنند. وقتی فاختگان بالغ در تمام یک محیط زندگی می کنند، مشکل می توان تشخیص داد که کدام فاخته به کدام گروه تعلق دارد. برای حل این مشکل، گروه بندی فاخته ها به روش خوشه بندی k-means انجام می شود (در شبیه سازی، مقدار 3-5 برای k کافی به نظر می رسد). اکنون که گروه های فاخته تشکیل شده اند، سود میانگین آن ها محاسبه می شود. مقدار بیشینه این میانگین ها گروه هدف را تعیین می کند. نتیجتاً، بهترین محل زندگی برای این گروه، مقصد جدید فاخته های مهاجر است.

در هنگام حرکت به سمت نقطه هدف، فاخته ها تمام طول راه به سمت مقصد را پرواز نمی کنند. آن ها فقط بخشی از راه را پرواز می کنند و همچنین یک انحراف از مسیر دارند. این شیوه حرکت به وضوح در شکل 3 نشان داده شده است. همان طور که در تصویر 3 دیده می شود، هر فاخته فقط $\lambda\%$ کل فاصله تا مقصدش را پرواز می کند و همچنین انحرافی

این مقاله، تنها تعداد N_{max} از فاخته ها که مقادیر سود بالاتری دارند، زنده می ماند، بقیه از بین می روند.

3.5 همگرایی

بعد از چند تکرار، همه جمعیت فاخته ها به بهترین محل زندگی با بیشترین شباهت به تخم های پرندگان میزبان و نیز بیشترین منابع غذا مهاجرت می کنند. این محل زندگی بیشترین سود ممکن را ایجاد می کند. کمترین تلفات تخم ها در این محل زندگی رخ می دهد. همگرایی بیش از 95٪ از فاخته ها به یک محل زندگی، پایان الگوریتم بهینه سازی فاخته خواهد بود. مراحل اصلی COA در تصویر 4 به شکل شبه کد نشان داده شده اند. در قسمت بعد، COA به چند مسئله بهینه سازی محک اعمال می شود.

اثبات های تئوری برای همگرایی به قوانین احتمال مجانبی در همه الگوریتم های بهینه سازی تصادفی، با توجه به طبیعت مارکوفی فرآیندهای زیربنا، نیاز به نوعی تعادل با جزئیات یا شرط برگشت پذیری دارد که به این معنی است که الگوریتم بخش بزرگی از کارایی ش را از دست می دهد. از این گذشته، اگر بر همگرایی نهایی به بهینگی سراسری با مفهوم قوی یا حتی ضعیفش، تاکید شود، نیاز به آنیلینگ خیلی آهسته ای هم هست. قوت الگوریتم های تصادفی از این حقیقت نشأت می گیرد که طبیعت احتمالی بالای آن ها به ما این اطمینان را می دهد که الگوریتم لزوما در بهینگی محلی گرفتار نمی شود و نیازی به استفاده از هنر اطلاعاتی روی گرادیان های هدف هم نیست، که باز هم

1. به محل زندگی فاخته ها با چند نقطه تصادفی روی تابع سود، مقدار بدهید.
2. به هر فاخته تعدادی تخم اختصاص دهید.
3. ELR را برای هر فاخته مشخص کنید.
4. به فاخته ها اجازه دهید در ELR مربوط به خود، تخم گذاری کنند.
5. تخم هایی که توسط پرندگان میزبان شناسایی می شوند را از بین ببرید.
6. اجازه دهید جوجه ها به دنیا بیایند و رشد کنند.
7. محل زندگی هر فاخته بالغ جدید را ارزیابی کنید.
8. تعداد حداکثر فاخته ها در محیط را محدود کنید و آنهایی که در بدترین محل ها زندگی می کنند را از بین ببرید.
9. فاخته ها را دسته بندی کنید و بهترین گروه را بیابید و محل زندگی هدف را برگزینید.
10. به جمعیت جدید فاخته ها اجازه مهاجرت به محل زندگی هدف بدهید.

شکل 4. شبه کد الگوریتم بهینه سازی فاخته

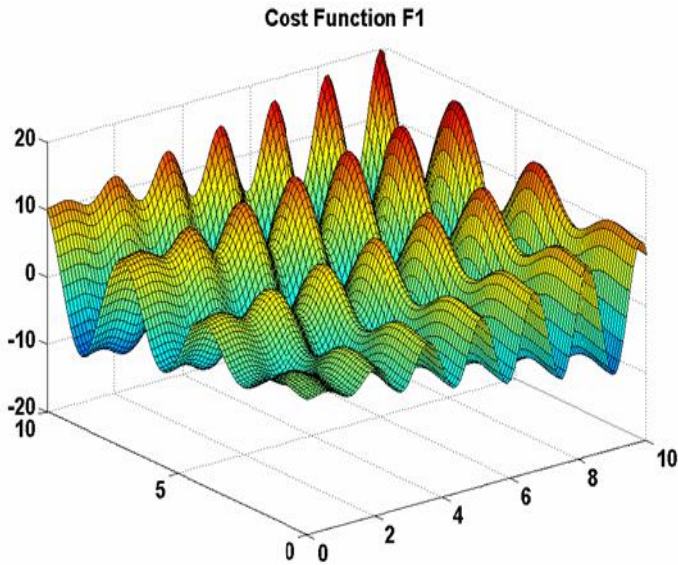
نیازی به استفاده از هر اطلاعاتی روی گرادیان های هدف هم نیست، که باز هم نیاز به شرایط تشخیص پذیری دارند.

4. محک های الگوریتم بهینه سازی فاخته

در این بخش، الگوریتم پیشنهاد شده (COA) با چهار تابع محک از منبع [1]، یک تابع راسریجین 10 بعدی و یک مطالعه موردی واقعی آزموده می شود.

4.1. توابع هزینه آزمون

تمام توابع محک مسائل مینیمم سازی هستند. این توابع در زیر آمده اند:



شکل 5. نمودار سه بعدی تابع F1

تابع F1:

$$f = x \times \sin(4x) + 1.1y \times \sin(2y)$$

$$0 < x, y < 10$$

$$\text{minimum: } f(9.039, 8.668) = -8.5547$$

تابع F2:

$$f = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2} - 0.5)}{1 + 0.1x^2 + y^2}$$

$$0 < x, y < 2, \text{ minimum: } f(0, 0.5) = 0.5$$

تابع F3:

$$f = (x^2 + y^2)^{0.25} \times \sin\{30[(x+0.5)^2 + y^2]^{0.1}\} + |x| + |y|$$

$$-\infty < x, y < +\infty$$

$$\text{minimum: } f(-0.2, 0) = -0.2471$$

تابع F4:

$$f = \frac{1}{10}(x^2 + y^2) + 0.1|1-x| + 0.1|1-y|$$

$$-\infty < x, y < +\infty$$

$$\text{minimum: } f(1, 1.6606) = -0.3356$$

تابع F5 (تابع راسریجین 10 بعدی):

$$f = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)), n = 9$$

$$-5.12 \leq x_i \leq 5.12$$

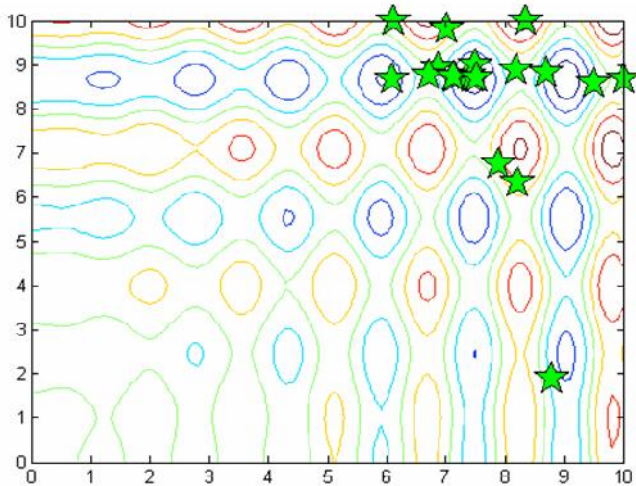
$$f(0, 0, \dots, 0) = 0$$

تابع اول F1 مورد بررسی قرار می گیرد. این تابع مینیموم -

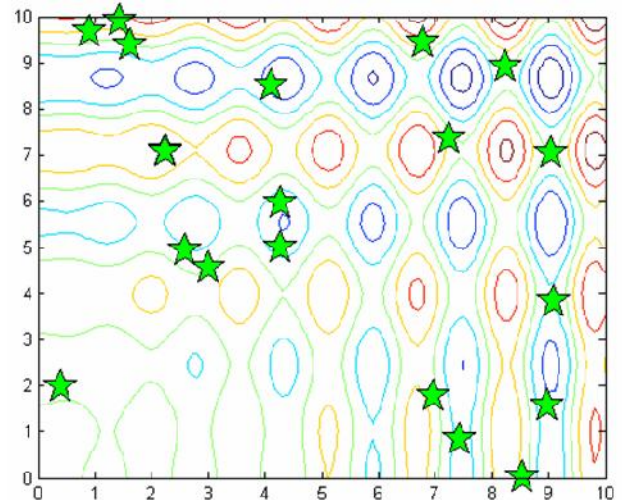
مطلقاً برابر 18.5547- در نقطه $(x, y) = (9.039, 8.668)$ در بازه $0 < x < 10$ و $0 < y < 10$ دارد. تصویر 5 نمودار سه بعدی این تابع را نشان می دهد.

تعداد اولیه فاخته ها تنها 20 تا در نظر گرفته می شود. هر فاخته می تواند بین 5 تا 10 تخم بگذارد. تصویر 6 پراکنندگی اولیه فاخته ها در محیط مسئله را نشان می دهد.

تصاویر 7 تا 12 محل زندگی جمعیت فاختگان را در دفعات پی در پی نشان می دهد. همگرایی در تکرار 7 رخ می دهد. COA تنها در 7 تکرار، مینیمم مطلق را به دست آورده است. همان طور که در تصاویر دیده می شود، فاخته ها در تکرار 4 دو نقطه کمینه دارند. سپس در تکرار 5 می بینیم که یک



شکل 8. محل زندگی فاخته ها در تکرار سوم

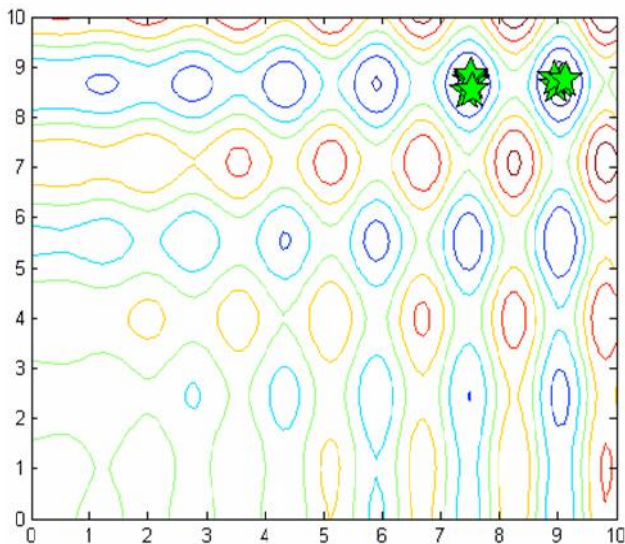


شکل 6. محل زندگی اولیه فاخته ها

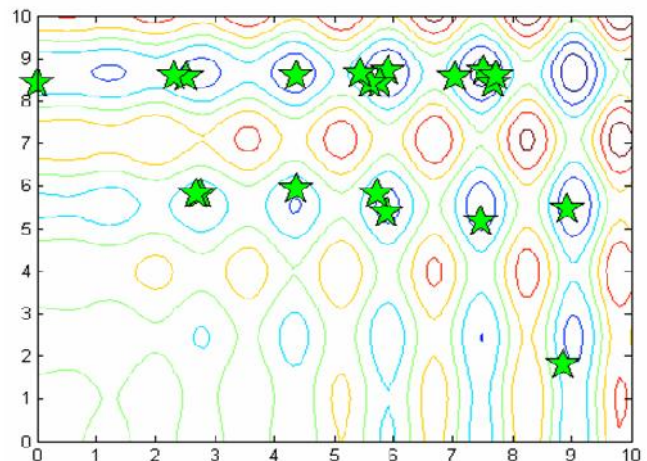
نشان می دهد.

برای انجام یک مقایسه، PSO و GA پیوسته با گزینش بر اساس چرخ رولت¹، تقاطع یکنواخت نیز در حل این تابع به کار گرفته می شوند. جمعیت اولیه در GA نیز 20 در نظر

گروه فاختگان به محل مینیمم مطلق مهاجرت می کند. در تکرار 6 اکثر فاخته ها در نقطه مینیمم مطلق هستند؛ و بالاخره در تکرار 7 تقریباً همه فاخته ها در بهترین محل زندگی به سر می برند، که همان مینیمم مطلق مسئله است. این محل زندگی (9.0396, 8.6706) با میزان هزینه 18.5543- است. تصویر 13 مینیمم سازی هزینه برای تابع آزمون 1 را

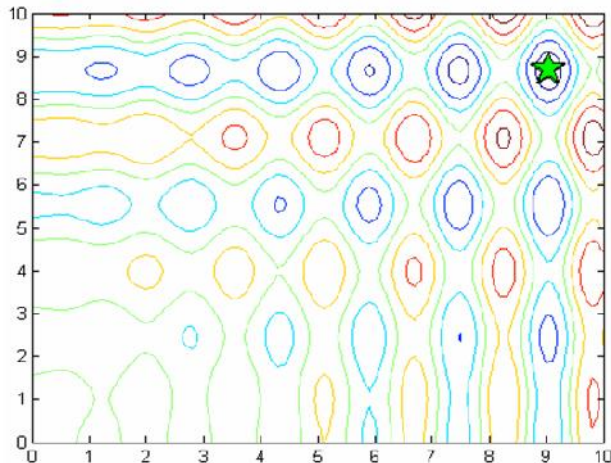


شکل 9. محل زندگی فاخته ها در تکرار چهارم

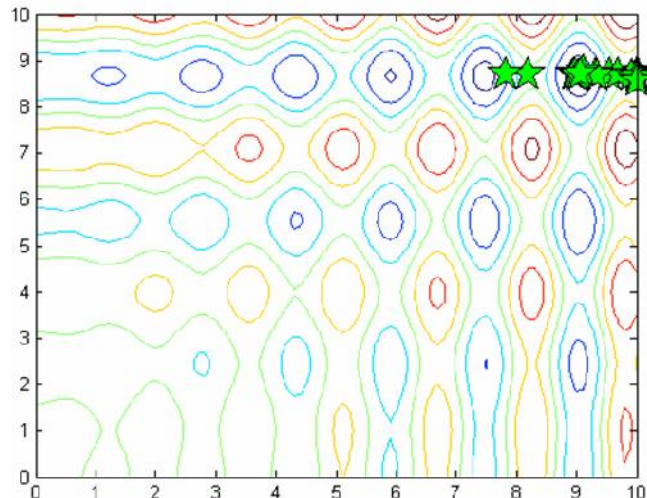


شکل 7. محل زندگی فاخته ها در تکرار دوم

¹ Roulette wheel



شکل 12. محل زندگی فاخته ها در تکرار هفتم



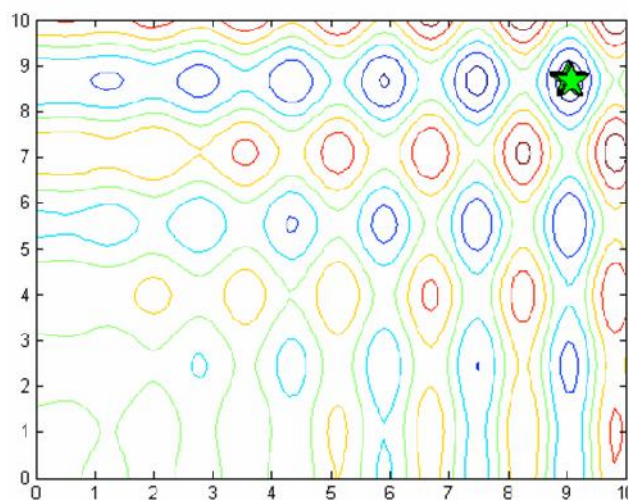
شکل 10. محل زندگی فاخته ها در تکرار پنجم

برای به دست آوردن میانگین انتظار از عملکرد هر روش انجام می شود. اجرای شبیه سازی برای 30 بار میانگینی به اندازه 45.9، 38.7 و 6.8 برای دفعات تکرار تا رسیدن به جواب در (به ترتیب) GA، PSO و COA به دست می دهد.

تصویر 14 نمونه یک نمودار کمینه سازی هزینه در تابع F1 برای GA در 100 تکرار را نشان می دهد.

همان طور که در تصویر 14 مشاهده می شود، GA در تکرار 24ام به مینیمم مطلق رسیده است. بهترین کروموزوم (9.0434, 8.6785) و مقدار هزینه 18.5513- است. تصویر 15 کمینه سازی تابع F1 را با استفاده از PSO نشان می دهد.

همان طور که در تصویر 15 دیده می شود، PSO در 19امین تکرار به مینیمم مطلق رسیده است. بهترین موقعیت ذره (9.0390, 8.6682) است، و هزینه در این نقطه برابر 18.5547- می باشد. با توجه به جدول 1 مشاهده می شود که در حالی که GA و PSO به میانگین دفعات تکرار 46.8 و 39.1 نیاز

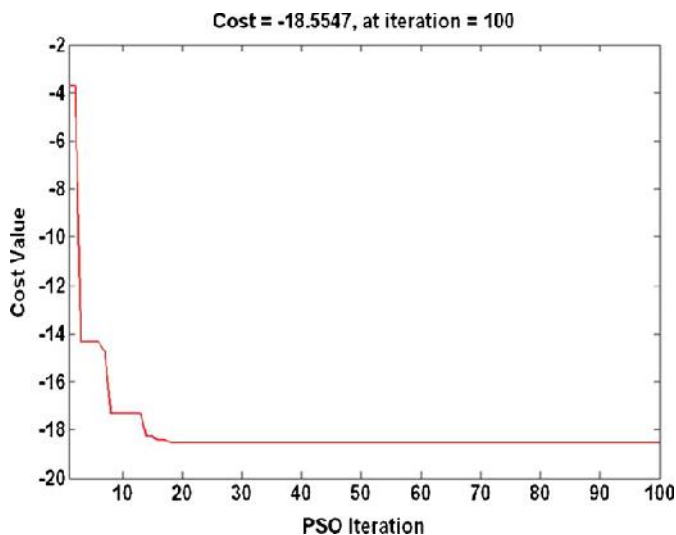


شکل 11. محل زندگی فاخته ها در تکرار ششم

گرفته می شود و نرخ های جهش و گزینش نیز به ترتیب 0.2 و 0.5 تعیین می شوند. برای PSO پارامتر ادراکی و پارامتر اجتماعی 2 در نظر گرفته می شوند. با توجه به این که جمعیت های اولیه متفاوت در هر روش مستقیماً بر نتیجه نهایی و سرعت الگوریتم اثر می گذارد، یک سری آزمون

تصاویر 16-18 نمودار کمینه سازی هزینه را برای هر سه الگوریتم برای توابع F1، F2 و F3 در یک اجرای تصادفی نشان می دهد. جدول 1 میانگین دفعات تکرار تا رسیدن به جواب را برای توابع آزمون ذکر شده، نشان می دهد.

دارند، COA به طور متوسط در 6.9 تکرار (تقریباً 7) به نقطه هدف دست می یابد. تا به اینجا می توان نتیجه گرفت که COA از GA و PSO بهتر عمل کرده است.



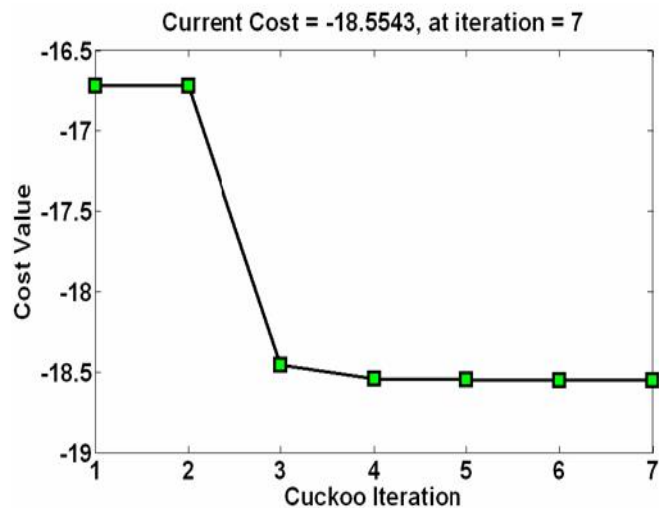
شکل 15. کمینه سازی هزینه با کمک PSO

	F2	F3	F4
GA	12.6	52.2	44.1
PSO	10.3	24.8	38.6
COA	5.2	6.9	6.3

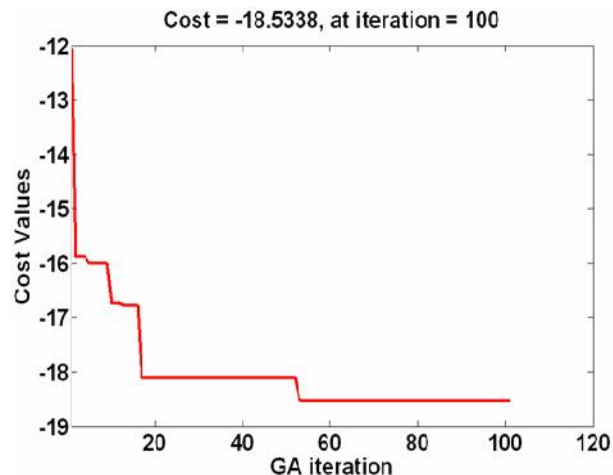
جدول 1. متوسط تکرار ها تا توقف برای GA، PSO و COA در 30 دوره

جالب ترین نکته در تصاویر 16-18 و همچنین جدول 1، «همگرایی سریع تر» الگوریتم بهینه سازی فاخته است.

با توجه به نتایج به دست آمده از توابع آزمون F1، F2، F3 و F4 ملاحظه می شود که هر سه روش قادر به یافتن مینیمم مطلق بوده اند. شاید تنها نکته جالب الگوریتم بهینه سازی

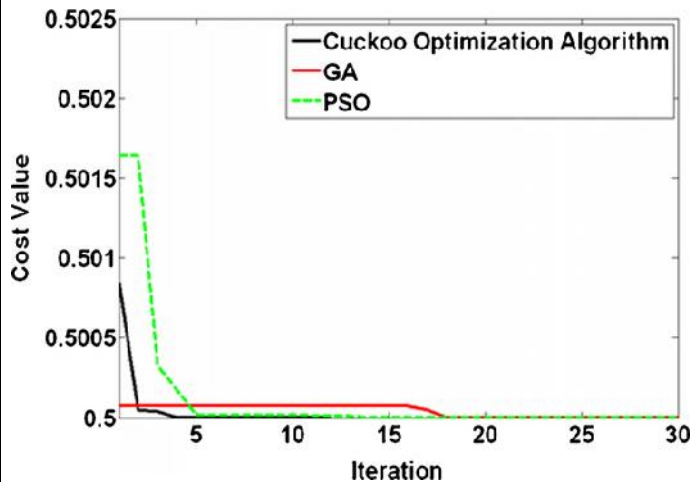


شکل 13. کمینه سازی هزینه برای تابع آزمون F1

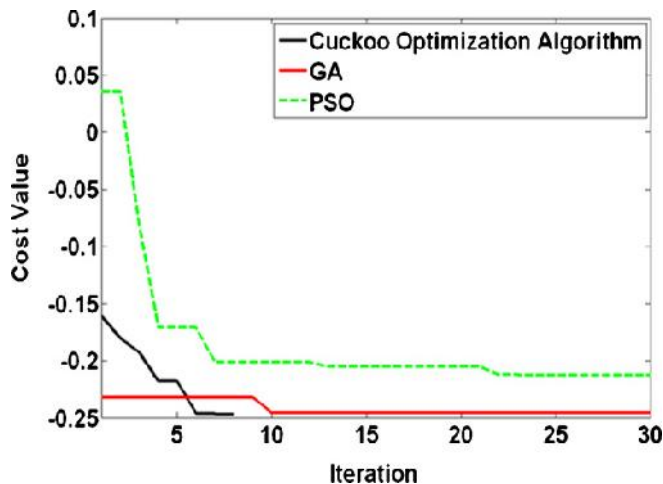


شکل 14. کمینه سازی هزینه با کمک GA

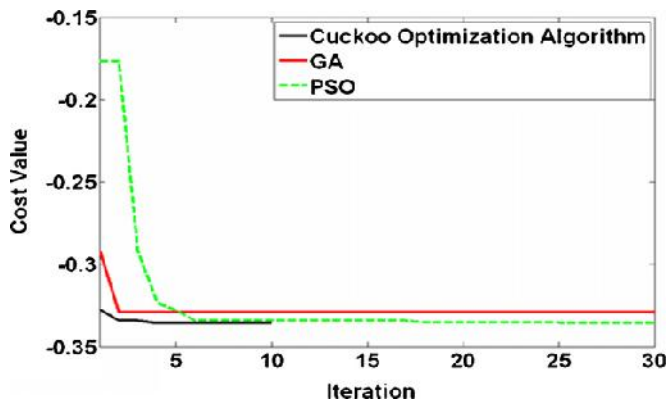
برای آزمون بیشتر، این سه الگوریتم بهینه سازی را روی توابع آزمون F1، F2 و F3 نیز پیاده می کنیم.



شکل 16. نمودار کمینه سازی هزینه برای تابع F2



شکل 17. نمودار کمینه سازی هزینه برای تابع F3



شکل 18. نمودار کمینه سازی هزینه برای تابع F4

فاخته (COA) همگرایی سریع تر آن باشد. اما برای نشان دادن برتری COA بر GA و PSO، تابع راسریجین 10 بعدی به عنوان تابع آزمون F5 انتخاب می شود. این تابع مینیمم های محلی متعددی دارد و حتی در سه بعد، یکی از مسائل دشوار است. شکل 19 تابع راسریجین 3 بعدی را نشان می دهد.

همان طور که حتی در سه بعد هم دیده می شود، تابع راسریجین یک مسئله بهینه سازی بسیار چالش برانگیز است. اما برای مشاهده عملکرد واقعی COA، GA و PSO، تابع راسریجین 10 بعدی به عنوان آخرین تابع محک برگزیده می شود. تصویر 20 نتایج کمینه سازی را برای هر سه الگوریتم نمایش می دهد. برای هر سه روش، اندازه اولیه جمعیت و بیشینه تعداد تکرارها به ترتیب 20 و 100 در نظر گرفته شده است.

حال، به روشنی می بینیم که GA و PSO نتوانسته اند در 100 تکرار به مینیمم مطلق دست یابند، ولی COA تنها در 66 تکرار به مینیمم مطلق تقریبی $f(x^*)=0$ همگرا شده است. در این تابع محک، COA به طور چشمگیری بهتر عمل کرده است و تخمین بسیار خوبی از مینیمم مطلق واقعی یافته است.

پس از آن که عملکرد فوق العاده COA در توابع هزینه آزمون به اثبات رسید، لازم است عملکرد آن را در مسائل

4.2. طراحی کنترل گر چند متغیره

4.2.1. کنترل گر PID برای فرآیندهای چند ورودی- چند

خروجی (MIMO)

در تصویر 21، حلقه کنترل PID چند متغیره را در نظر بگیرید.

فرآیند چند متغیره $P(s)$ در تصویر 21 می تواند به شکل زیر نمایش داده شود:

$$P(s) = \begin{bmatrix} p_{11}(s) & \dots & p_{1n}(s) \\ \vdots & \ddots & \vdots \\ p_{n1}(s) & \dots & p_{nn}(s) \end{bmatrix}$$

که $p_{ij}(s)$ تابع تبدیل بین y_i و u_j است. در تصویر 21، بردارهای Y_d ، Y ، U و E به شکل زیر هستند:

$$Y_d = [y_{d1} \ y_{d2} \ \dots \ y_{dn}]^T$$

$$Y = [y_1 \ y_2 \ \dots \ y_n]^T$$

$$U = [u_1 \ u_2 \ \dots \ u_n]^T$$

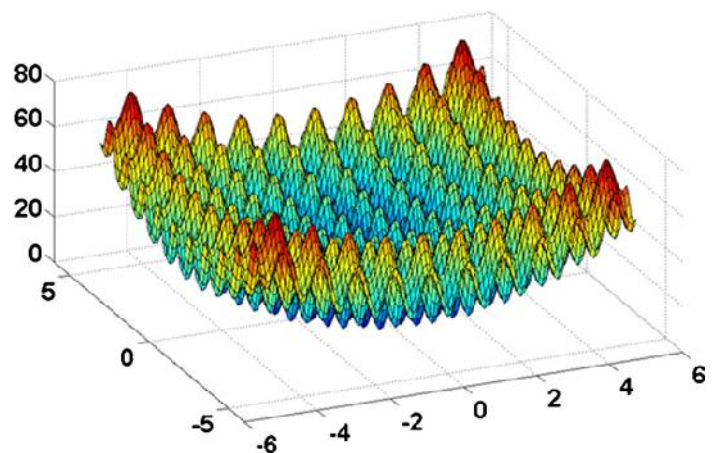
$$E = Y_d - Y = [e_{11} \ e_{22} \ \dots \ e_{nn}]^T$$

کنترل گر چند متغیره PID، $C(s)$ در تصویر 21، به شکل زیر است:

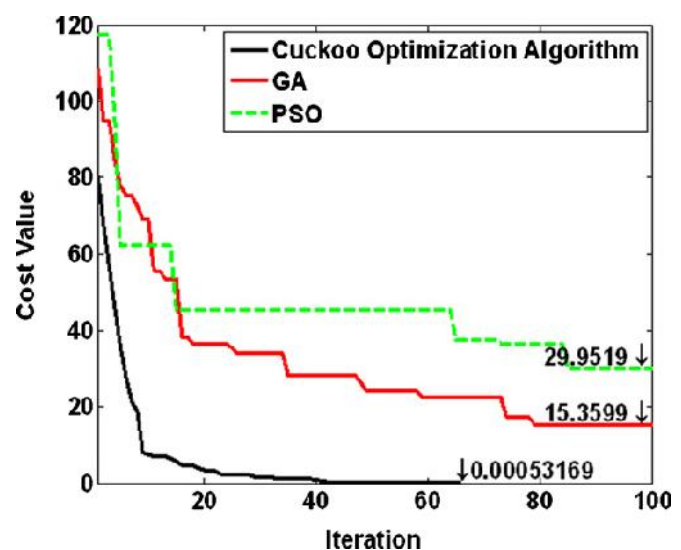
$$C(s) = \begin{bmatrix} c_{11}(s) & \dots & c_{1n}(s) \\ \vdots & \ddots & \vdots \\ c_{n1}(s) & \dots & c_{nn}(s) \end{bmatrix}$$

$c_{ij}(s)$ که $i, j \in \{1, 2, \dots, n\}$ به صورت زیر است:

واقعی بررسی کنیم. به این منظور، فرآیند برج تقطیر چند ورودی- چند خروجی (MIMO) انتخاب می شود تا به وسیله یک کنترل گر PID چند متغیره کنترل شود. پارامترهای کنترل گر PID با استفاده از COA، GA و روش پیشنهاد شده در [29] طراحی می شوند. قبل از نشان دادن فرآیند طراحی، توضیح مختصری درباره طراحی کنترل گر چند متغیره داده می شود.



شکل 19. نمودار سه بعدی تابع راسریجین



شکل 20. کمینه سازی هزینه برای تابع راسریجین 10 بعدی

با توجه به هدف جداسازی، IAE به شکل زیر تعریف می شود:

$$IAE \triangleq \sum_{i=1}^n \sum_{j=1}^n IAE_{ij} \triangleq \sum_{i=1}^n \sum_{j=1}^n \int_0^{\infty} (|e_{ij}(t)|) dt$$

که $|e_{ii}(t)|$ قدر مطلق خطای خروجی $y_i(t)$ در هنگام پیگیری ورودی $y_{di}(t)$ است و $|e_{ij}(t)|$ قدر مطلق خطای به وجود آمده توسط تاثیر ورودی $y_{dj}(t)$ روی خروجی $y_i(t)$ است ($i \neq j$). همچنین IAE_{ij} انتگرال قدر مطلق خطای $e_{ij}(t)$ در طی زمان است. منبع $|e_{ij}(t)|$ همان مسئله جفتگری است.

شرط دیگر اجرا که در طراحی کنترل گر استفاده می شود، درصد فرجهش^۷ (PO) و درصد فروجهش^۸ (PU) است به شکل زیر تعریف می شود:

$$POU \triangleq \sum_{i=1}^n \sum_{j=1}^n POU_{ij} \triangleq \sum_{i=1}^n \sum_{j=1}^n \text{Max}\{PO_{ij}, PU_{ij}\}$$

هدف، طراحی یک کنترل گر به منظور پیگیری خروجی های مطلوب و جداسازی فرآیند کنترل شده تا حد ممکن است. به خاطر این که POU معمولاً مقادیر کمی نسبت به IAE دارد و نیز برای تاکید بر POU، ما 10 برابر POU را به IAE می افزاییم تا تابع هدف را بسازیم.

$$\text{Cost} = IAE + 10 \times POU$$

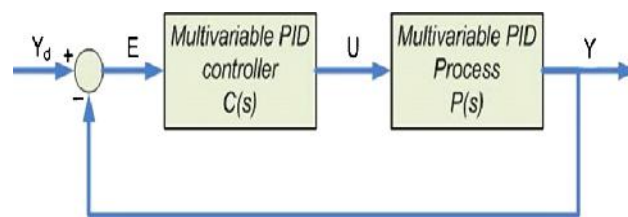
$$c_{ij}(s) = K_{Pij} + K_{Iij} 1/s + K_{Dij}s$$

K_{Pij} تناسب، K_{Iij} ، انتگرال و K_{Dij} مشتق کنترل گر PID، $c_{ij}(s)$ هستند.

4.2.2 طراحی تکاملی PID

در طراحی کنترل گر های PID، هدف تنظیم ضرایب K_p ، K_i و K_d است تا خروجی برخی مشخصات مطلوب را داشته باشد. معمولاً در حیطه زمان، این مشخصات با عناوین فرجهش^۱، زمان خیز^۲، زمان نشست^۳ و خطای حالت دائم^۴ مشخص می شوند. دو نوع از شروط اجرا در پیگیری خروجی که معمولاً در طراحی کنترل گر در نظر گرفته می شوند، انتگرال مربع خطا^۵ (ISE) و انتگرال قدر مطلق خطا^۶ (IAE) در خروجی مطلوب است.

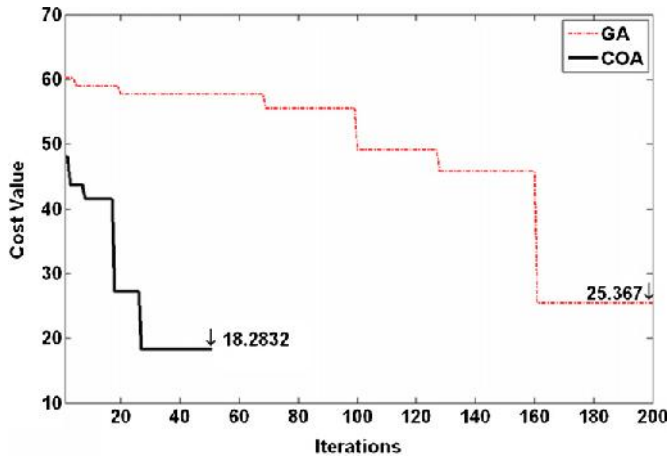
در طراحی یک کنترل گر چند متغیره، یکی از اهداف اصلی، غالب قطری بودن فرآیند کنترل است. این بدان معناست که کنترل گر به گونه ای طراحی شود که $y_i(t)$ بتواند ورودی مطلوب $y_{di}(t)$ را پیگیری کند و پاسخ سایر ورودی ها $y_{dj}(t)$ که $j \in \{1, 2, \dots, n \mid i \neq j\}$ را رد کند.



شکل 21. نمودار بلوکی یک فرآیند کنترل شده چند متغیره

⁵ Integral Squared Error
⁶ Integral Absolute Error
⁷ Percentage of Overshoot
⁸ Percentage of Undershoot

¹ overshoot
² rise time
³ settling time
⁴ steady state error

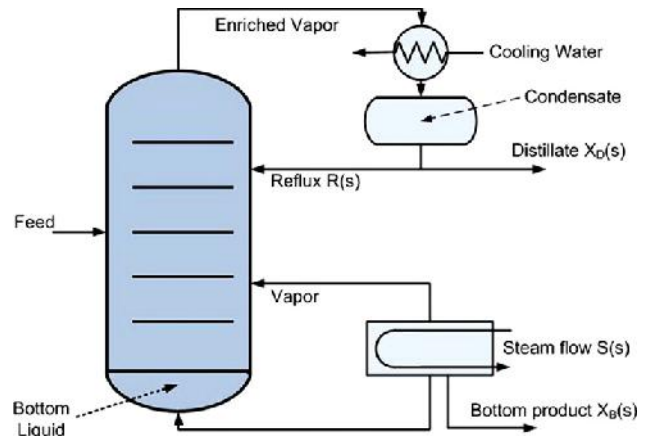


شکل 23. هزینه حداقل COA و GA

فرآیند تقطیر، و درصد متانول در محصول زیرین برج می باشد. همچنین $R(s)$ و $S(s)$ به ترتیب، نرخ جریان برگشت (ریفلاکس) و نرخ جریان بخار در ریویولر است.

DCS یک سیستم چند ورودی-چند خروجی (MIMO) 2×2 با تعاملات قوی میان ورودی ها و خروجی ها است. چهار تابع تبدیل در فرآیند چند متغیره، دینامیک مرتبه اول و تاخیرهای زمانی قابل توجه دارند. اهداف کنترل الف) پیگیری ورودی های کنترل y_{1d} و y_{2d} با ورودی y_1 و y_2 و ب) غالب قطری بودن فرآیند تحت کنترل تا حد ممکن می باشد. در [29] یک کنترل گر PID چند متغیره برای DCS با استفاده از روش رله فید بک متمرکز (DRF) طراحی می شود. عناصر قطری و غیر قطری این کنترل گر به ترتیب در دو فرم PI و PID طراحی می شوند. این کنترل گر به صورت زیر است:

پارامترهای کنترل گر PID، با استفاده از الگوریتم تکاملی بهینه سازی پیشنهاد شده، برای نوعی از برج تقطیر تنظیم می شوند. نتایج به دست آمده توسط این روش با نتایج به دست آمده از الگوریتم ژنتیک (GA) و روش معرفی شده در [29] به نام رله فیدبک نامتمرکز¹ (DRF) مقایسه می شود. 4.2.2.1 مطالعه موردی تجربی: در این جا یک کنترل گر PID چند متغیره برای یک سیستم شیمیایی MIMO طراحی می شود. این سیستم یک نوع مدل 2×2 برج تقطیر است [30]. یک شمای ساده از سیستم برج تقطیر² (DCS) در تصویر 22 نشان داده می شود.



شکل 22. شمای ساده ای از سیستم ستون تقطیر

تابع تبدیل ماتریس DCS به صورت زیر است:

$$\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8 e^{-s}}{1 + 16.7s} & \frac{-18.9 e^{-3s}}{1 + 21s} \\ \frac{6.6 e^{-7s}}{1 + 10.9s} & \frac{-19.4 e^{-3s}}{1 + 14.4s} \end{bmatrix} \cdot \begin{bmatrix} R(s) \\ S(s) \end{bmatrix}$$

که $X_D(s)$ و $X_B(s)$ به ترتیب درصد متانول در محصول -

² Distillation Column System

¹ Decentralized Relay Feedback

ها 200 در نظر گرفته می شود. اما COA در 52 تکرار به مقدار هزینه کل 18.28 می رسد و الگوریتم متوقف می شود.

از GA با جمعیت اولیه 100، انتخاب تورنومنت، حذف گوسی و تقاطع پراکنده برای تنظیم پارامترهای کنترل گر چند متغیره PID برای فرآیند استفاده می شود. برای بهره برداری کامل از پتانسیل GA در مینیم سازی هزینه، آن را به یک تابع هیبرید مجهز کرده ایم.

تصویر 23 هزینه های حداقل را برای بهترین نتایج از 10 اجرای متفاوت COA و GA نشان می دهد. همان طور که در این تصویر دیده می شود، میزان همگرایی حالت دائم COA برابر 18.283 است، از همین میزان در GA، یعنی 25.367 کمتر می باشد.

پارامترهای کنترل گر PID و مقادیر هزینه مربوط به آن ها، که توسط COA، GA و DRF به دست آمده است؛ در جداول 2 و 3 نمایش داده شده است. بنابر جدول 3

Criteria	Method		
	COA	GA	DRF
IAE ₁₁	3.849	6.5688	4.9278
IAE ₁₂	0.99608	1.1735	1.0625
IAE ₂₁	2.6465	3.8051	4.4716
IAE ₂₂	7.1225	7.2117	9.0288
IAE	14.614	18.759	19.4907
POU ₁₁	9.1797%	16.118%	9.91%
POU ₁₂	7.2331%	8.8628%	4.07%
POU ₂₁	10.967%	31.293%	22.05%
POU ₂₂	9.3256%	9.8398%	9.86%
POU	36.705%	66.114%	45.89%
Cost	18.283	25.376	24.0791

جدول 3 قسمت های مختلف تابع هزینه، بهینه شده توسط روش های COA، GA و DRF

$$C(s) = \begin{bmatrix} 0.184 + 0.0469 \frac{1}{s} & -0.0102 - 0.0229 \frac{1}{s} + 0.0082s \\ -0.0674 + 0.0159 \frac{1}{s} - 0.0536s & -0.066 - 0.0155 \frac{1}{s} \end{bmatrix}$$

به منظور مقایسه نتایج COA و GA با روش DRF در تنظیم پارامترهای کنترل گر PID برای وسیله تعریف شده در (17)، کنترل گر C(s) به شکل زیر در نظر گرفته می شود:

$$C(s) = \begin{bmatrix} K_{P11} + K_{I11} \frac{1}{s} & K_{P12} + K_{I12} \frac{1}{s} + K_{D12}s \\ K_{P21} + K_{I21} \frac{1}{s} + K_{D21}s & K_{P22} + K_{I22} \frac{1}{s} \end{bmatrix}$$

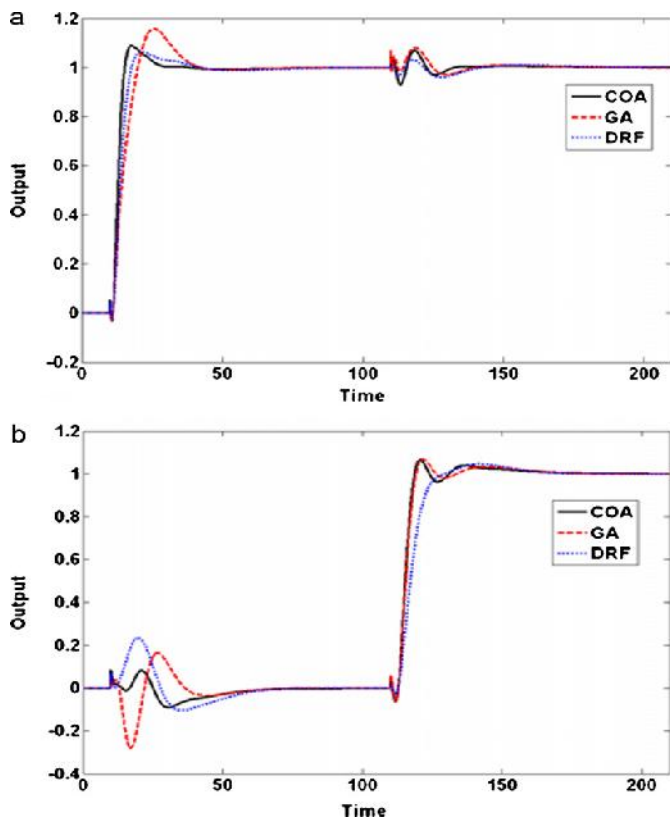
بنابراین هدف، حل یک مسئله بهینه سازی 10 بعدی برای تعیین ضرایب بهینه

$[K_{P11} \ K_{I11} \ K_{P12} \ K_{I12} \ K_{D12} \ K_{P21} \ K_{I21} \ K_{D21} \ K_{P22} \ K_{I22}]$ به منظور کمینه کردن تابع هزینه (16) خواهد بود. COA و GA 10 بار در این مسئله به کار گرفته می شوند و بهترین نتیجه هر کدام در این بخش مورد مطالعه قرار می گیرد.

PID parameters	Method		
	COA	GA	DRF
K _{P11}	0.2751	0.1763	0.184
K _{I11}	0.0803	0.0592	0.0469
K _{P12}	-0.0675	-0.0418	-0.0102
K _{I12}	-0.0290	-0.0246	-0.0229
K _{D12}	0.0835	0.1037	0.0082
K _{P21}	-0.0522	0.0404	-0.0673
K _{I21}	0.0330	0.0227	0.0159
K _{D21}	-0.0680	-0.0425	-0.0536
K _{P22}	-0.1243	-0.1127	-0.066
K _{I22}	-0.0210	-0.019	-0.0155

جدول 2. پارامترهای کنترل گر PID به دست آمده از COA، GA و DRF

از COA با جمعیت اولیه ای برابر 20 فاخته و بیشینه تعداد تخم برابر 5 برای تنظیم پارامترهای کنترل گر استفاده می شود. هم برای COA و هم برای GA، حداکثر تعداد تکرار



شکل 24. پاسخ فرآیند برج تقطیر به تأخیرهای مختلف در ورودی های پله ای: (a) اولین خروجی، (b) دومین خروجی

انحراف پدید آمده هنگام حرکت به سوی محل زندگی هدف، باعث می شود جمعیت فاخترگان نسبت به این که مستقیماً در یک خط حرکت کنند؛ مناطق بیشتری را جست و جو کنند. پس از چند مهاجرت، همه جمعیت فاخته ها در یک محل زندگی، که بهترین موقعیت در منطقه است، جمع می شوند. الگوریتم معرفی شده روی 5 تابع هزینه محکم آزموده شد. مقایسه COA با نسخه های استاندارد PSO و GA با انتخاب چرخ رولت و تقاطع یکنواخت، برتری COA را در همگرایی سریع و دست یابی به نقطه بهینه سراسری نشان داد. در 4 تابع آزمون اول، همه روش ها مینیمم مطلق را یافتند، ولی COA سریع تر و در تکرارهای کم تر همگرا شد. ولی در آخرین تابع آزمون

کنترل گر به دست آمده توسط DRF تنها باعث کمترین POU₁₂ شده است. با توجه به سایر پارامترها، به روشنی دیده می شود که کنترل گر طراحی شده توسط COA بهترین کنترل گر طراحی شده توسط این سه روش است. مقادیر جدول 3 نشان می دهد که با استفاده از کنترل گر با پارامترهای طراحی شده توسط COA، هر دو خروجی بهترین پیگیری و کمترین اتصال را دارند. هزینه کل به دست آمده در جدول 3 عملکرد بهتر کنترل گر طراحی شده توسط COA را نشان می دهد.

تصویر 24 پاسخ فرآیند برج تقطیر کنترل شده را به ورودی های پله ای با استفاده از کنترل گر های به دست آمده از COA، GA و DRF را نشان می دهد. برای داشتن نمایی بهتر از جداسازی به وجود آمده توسط کنترل گر های مختلف، ورودی های پله ای با تاخیرهای زمانی 110 ثانیه ای اعمال شده اند.

5. نتیجه گیری

در این مقاله، یک الگوریتم بهینه سازی جدید پیشنهاد شد که از زندگی یک پرنده به نام فاخته الهام گرفته است. ویژگی های منحصر به فرد فاخته ها در تخم گذاری و تکثیر انگیزه اصلی ایجاد این الگوریتم بهینه سازی جدید بوده است. هر فاخته در این الگوریتم یک محل زندگی دارد که در گوشه و کنار آن شروع به تخم گذاری می کند. اگر تخم ها زنده بمانند، بزرگ می شوند و تبدیل به فاخته بالغ می شوند. سپس به منظور تکثیر، فاخته ها به بهترین محل زندگی که تا به حال یافت شده، مهاجرت می کنند.

- [13] V. Chellaboina, M.K. Ranga, Reduced order optimal control using genetic algorithms, in: American Control Conference, Portland, USA, vol. 2, 2005, pp. 1407–1412.
- [14] D.B. Fogel, An evolutionary approach to the traveling salesman problem, *Bio- logical Cybernetics* 60 (2004) 139–144.
- [15] B. Bontoux, D. Feillet, Ant colony optimization for the traveling purchaser problem, *Computers & Operations Research* 35 (2008) 628–637.
- [16] R.L. Johnston, H.M. Cartwright, *Applications of Evolutionary Computation in Chemistry*, Springer-Verlag, Berlin, Heidelberg, 2004.
- [17] C.-C. Wu, K.-C. Lai, R.-Y. Sun, GA-based job scheduling strategies for fault tolerant grid systems, in: *IEEE Asia-Pacific Services Computing Conference*, 2008, pp. 27–32.
- [18] L.V. Arruda, M.C. Swiech, M.R. Delgado, F. Neves Jr., PID control of MIMO process based on rank niching genetic algorithm, *Applied Intelligence* 29 (2008) 290–305.
- [19] H.A. Varol, Z. Bingul, A new PID tuning technique using ant algorithm, in: *Proceeding of the IEEE 2004 American Control Conference*, vol. 3, 2004, pp. 2154–2159.
- [20] H.H. Balaci, J.F. Valenzuela, Scheduling electric power generators using particle swarm optimization combined with Lagrangian relaxation method, *International Journal of Applied Mathematics and Computer Science* 14 (2004) 411–421.
- [21] C. Andrieu, A. Doucet, Simulated annealing for maximum a posteriori parameter estimation of hidden Markov models, *IEEE Transactions on Information Theory* 46 (2000) 994–1004.
- [22] *The Life of Birds, Parenthood.*
<http://www.pbs.org/lifeofbirds/home/index.html>
 (retrieved 05.09.09).
- [23] C. Ellis, C. Kepler, A. Kepler, K. Teebaki, Occurrence of the longtailed cuckoo
Eudynamis Taitensis on Caroline Atoll, Kiribati 90 (1990) 202.
- [24] P. Hockey, Patterns and correlates of bird migrations in Sub-Saharan Africa, *Emu* 100 (2000) 401–417.
- [25] Payne, B. Robert, *The Cuckoos*, Oxford University Press, 2005.
- [26] <http://www.telegraph.co.uk/earth/wildlife/4109282/Cuckoo-chicks-dupe-foster-parents-from-the-moment-they-hatch.html>.
- [27] N.A. Campbell, Fixed action patterns, in: *Biology*, fourth ed., Benjamin Cummings, New York, 1996, p. 1179.
- [28] D. Wheatcroft, Co-evolution: a behavioral 'spam filter' to prevent nest parasitism, *Current Biology* 19 (2009) R170–R171.
- [29] Q.G. Wang, B. Zou, T.H. Lee, B. Qiang, Auto-tuning of multivariable PID controllers from decentralized relay feedback, *Automatica* 33 (1997) 319–330.
- [30] W.L. Luyben, A simple method for tuning SISO controllers in a multivariable system, *Industrial and Engineering Chemistry Product Research and Development* 25 (1986) 654–660.

تابع راسریجین 10 بعدی) و GA و PSO نتوانستند حتی به مقداری نزدیک نقطه بهینه سراسری همگرا شوند. ولی COA تنها در 66 تکرار، تخمین بسیار خوب و قابل قبولی از مینیمم مطلق پیدا کرد. البته، باید توجه کرد که عملکرد بهتر COA در دست یابی به نتایج بهتر برای این پنج تابع محک نتیجه می دهد و یک مطالعه موردی واقعی لزوماً بدین معنا نیست که COA بهترین روش تکاملی به وجود آمده تا به حال است. تنها می توان آن را به عنوان یک تقلید موفق از طبیعت به حساب آورد؛ که برای برخی از انواع مسائل بهینه سازی مناسب است.

منابع

- [1] R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*, second ed., John Wiley & Sons, New Jersey, 2004.
- [2] M. Melanie, *An Introduction to Genetic Algorithms*, MIT Press, Massachusetts, 1999.
- [3] S.N. Sivanandam, S.N. Deepa, *Introduction to Genetic Algorithms*, Springer- Verlag, Berlin, Heidelberg, 2008.
- [4] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [5] A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, New Jersey, 2005.
- [6] M. Dorigo, C. Blum, Ant colony optimization theory: a survey, *Theoretical Computer Science* 344 (2005) 243–278.
- [7] M. Dorigo, L.M. Gambardella, Ant Colony System: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation* 1 (1997) 53–66.
- [8] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science: New Series* 220 (1983) 671–680.
- [9] J. De Vicente, J. Lanchares, R. Hermida, Placement by thermodynamic simulated annealing, *Physics Letters A* 317 (2003) 415–423.
- [10] G. Zilong, W. Sun'an, Z. Jian, A novel immune evolutionary algorithm incorporating chaos optimization, *Pattern Recognition Letters* 27 (2006) 2–8.
- [11] A. Ahrari, M. Shariat-Panahi, A.A. Atai, G.E.M.: a novel evolutionary optimization method with improved neighborhood search, *Applied Mathematics and Computation* 210 (2009) 376–386.
- [12] Lu Hong, A novel particle swarm optimization method using clonal selection algorithm, in: *International Conference on Measuring Technology and Mechatronics Automation*, vol. 2, 2009, pp. 471–474.