

ارائه یک الگوریتم جهش قورباغه بهبود یافته با مولفه شناختی

^۱ سمیه رنجکش، دانشکده برق، رایانه و فناوری اطلاعات دانشگاه آزاد اسلامی قزوین، s.ranjakesh@yahoo.com

^۲ محمدرضا میبیدی، عضو هیئت علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، mmeybodi@aut.ac.ir

^۳ بهروز معصومی، عضو هیئت علمی دانشکده برق-رایانه دانشگاه آزاد اسلامی واحد قزوین، masoumi_b@yahoo.com

چکیده

تکنیک‌های جستجو و بهینه‌سازی هستند که قاعده‌ی کلی آنها از سیستم‌های موجود در طبیعت الهام گرفته شده است (تکامل داروین). در این الگوریتم‌ها از فلسفه‌ی انتخاب اصلح در طبیعت، استفاده می‌شود و با الهام از طبیعت، پاسخ‌های یک مساله‌ی مفروض در کروموزوم‌ها کد می‌شود. تکامل کروموزوم‌ها که در نتیجه‌ی اعمالی مانند crossover، mutation و انتخاب طبیعی^۵ رخ می‌دهد از طریق کدهای کامپیوتری شبیه‌سازی می‌شود. تاکنون مشخص شده که EAها به تنهایی برای تنظیم دقیق جستجو در فضاها ترکیبی پیچیده مناسب نیستند و ترکیب آنها با تکنیک‌های دیگر، می‌تواند به مقدار زیادی کارایی جستجو را افزایش دهد. در [۱] ترکیب EAها با جستجوی محلی، به عنوان الگوریتم‌های ممتیک (MAS)^۶ نامگذاری شد. MAها از مدل‌های وفق‌پذیری سیستم‌های طبیعی که وفق تکاملی یک جمعیت را با یادگیری فردی اعضای جمعیت ترکیب می‌کنند، الهام گرفته می‌شوند. نام این الگوریتم از مفهوم مم (meme) در نظریه‌ی داروین، الهام گرفته شده است. مم، واحد سنجش تکامل فرهنگی است که می‌تواند مقدار اصلاح فرهنگی محلی صورت گرفته را منعکس کند. در حوزه‌ی بهینه‌سازی هیوریستیک، یک مم استراتژی یادگیری یا توسعه را نمایش می‌دهد. از این‌رو یک مدل ممتیک وفق‌پذیری، میزان انعطاف افراد جمعیت را نشان می‌دهد که مدل ژنتیک از نمایش آن معذور است.

MAها در [۲] نشان داده‌اند که در برخی از حوزه‌ها از نقطه نظر بهینه‌سازی، کارا تر و موثر تر از EAهای سنتی هستند. در نتیجه، MAها درجه پذیرش بالایی را بدست می‌آورند، به ویژه در حیطه‌ی مسائل بهینه‌سازی ترکیبی که نمونه‌های بزرگی از این مسائل توسط آنها بهینه‌سازی شده‌اند و سایر روش‌های متاهوریستیک نتوانسته‌اند نتایج قابل‌مقایسه‌ای را ایجاد کنند. SFL، عضو جدید خانواده‌ی الگوریتم‌های ممتیک، یک الگوی جستجوی جمعی مبتنی بر جمعیت است که از ممتیک‌های طبیعی الهام گرفته شده است. این الگوریتم از نحوه‌ی شکار غذای قورباغه‌ها الهام گرفته شده است. پژوهشگران دریافتند که حداقل از نقطه نظر تئوری، اعضای منفرد گروه در طول جستجوی غذا می‌توانند

جهش ترکیبی قورباغه (SFL)^۱ یک الگوی جستجوی جمعی مبتنی بر جمعیت است که از ممتیک‌های طبیعی الهام گرفته شده است. قابلیت تطبیق این الگوریتم با محیط‌های پویا، SFL را تبدیل به یکی از مهمترین الگوریتم‌های ممتیک کرده است. در این مقاله به منظور بهبود ثبات این الگوریتم و همچنین برای افزایش توانایی آن برای جستجوی بهینه‌ی سراسری، یک "مولفه‌ی شناختی"^۲ به SFL اضافه می‌شود. با وجود این مولفه، هر قورباغه نه تنها موقعیت خود را بر اساس بهترین عضو memplex یا بر اساس بهترین عضو موجود در جمعیت تنظیم می‌کند، بلکه در این میان موقعیت خودش را نیز می‌سنجد. برای ارزیابی SFL بهبود یافته (ISFL)^۳، شبیه‌سازی‌های مختلفی انجام گرفت و در آنها SFL و ISFL با استفاده از شش بنچمارک بهینه‌سازی گسسته و پیوسته مقایسه شدند. بر طبق نتایج شبیه‌سازی، افزودن رفتار شناختی به SFL به مقدار قابل توجهی کارایی این الگوریتم را برای حل مسائل بهینه‌سازی افزایش می‌دهد. این بهبود کارایی، با افزایش مقیاس مساله آشکارتر می‌شود.

کلمات کلیدی

الگوریتم جهش قورباغه، الگوریتم ممتیک، رفتار شناختی

Frog Leaping Algorithm, Memetic Algorithm, Cognition Component

۱. مقدمه

در طی سال‌های گذشته تاکنون، دانشمندان هنگام حل مسائل پیچیده، در طبیعت به دنبال راه حل گشته‌اند. الگوریتم‌های تکاملی (EAs)^۴ گروهی از

^۱ Shuffled frog leaping

^۲ 'cognition component'

^۳ improved SFL

^۴ Evolutionary Algorithms

^۵ natural selection

^۶ memetic algorithms

۲. مروری بر الگوریتم ترکیبی جهش قورباغه

الگوریتم SFL یک الگوریتم ممتیک متاهوریستک است که با این هدف طراحی شده است که راه حل بهینه‌ی سراسری را از طریق جستجوی هیوریستیک مطلع که بوسیله‌ی یک تابع هیوریستیک انجام می‌شود، بیابد. این الگوریتم مبتنی بر تکامل م‌هایی که توسط افراد تعاملی حمل می‌شوند و مبادله‌ی سراسری اطلاعات میان جمعیت است.

در طول الگوریتم SFL، قورباغه‌ها در اثر تکامل ممتیک تغییر شکل می‌یابند. در این الگوریتم، قورباغه‌ها به عنوان میزبانانی برای م‌ها تلقی می‌شوند و به صورت بردار ممتیک نمایش داده می‌شوند. هر م شامل تعدادی memotype است. Memotype‌ها یک عقیده را نمایش می‌دهند، درست مانند ژن‌ها در الگوریتم ژنتیک که یک ویژگی را در کروموزوم نمایش می‌دهند. SFL مشخصه‌های فیزیکی قورباغه‌های موجود در جمعیت را تغییر نمی‌دهد بلکه به طور پیش‌رونده عقیده‌های آنها را اصلاح می‌کند. قورباغه‌ها می‌توانند اطلاعاتشان را با یکدیگر مبادله کنند و از این طریق م‌هایشان را اصلاح کنند. در اثر بهبود م‌ها، میزان جهش هر قورباغه تنظیم می‌شود و بدین ترتیب موقعیت هر قورباغه تغییر می‌کند. بنابراین الگوریتم SFL از PSO به عنوان ابزاری برای جستجوی محلی و انجام رقابت و از الگوریتم SCE برای ترکیب اطلاعات حاصل از جستجوهای محلی و حرکت به سوی راه حل سراسری استفاده می‌کند.

نمونه‌ای از قورباغه‌های مجازی یک جمعیت را تشکیل می‌دهند. جمعیت به زیرگروه‌هایی به نام memplex تقسیم می‌شود. هر memplex می‌تواند به عنوان مجموعه‌ای متشکل از قورباغه‌هایی با فرهنگ مشابه تلقی کرد که در پی هدف یکسان هستند. قورباغه‌های موجود در memplex‌ها به صورت همزمان اطلاعاتشان را با یکدیگر مبادله می‌کنند و بدین ترتیب به سوی هدف مشترکشان پیش می‌روند. جهش قورباغه، م‌ی که توسط هر فرد حمل می‌شود را بهبود می‌بخشد و کارایی آن فرد را برای رسیدن به هدف افزایش می‌دهد. قورباغه‌های مفرد موجود در هر memplex می‌توانند تحت تاثیر سایر قورباغه‌ها قرار گیرند و بدین ترتیب دچار تکامل ممتیک می‌شوند. بعد از اینکه مرحله‌ی تکامل ممتیک به تعداد معینی تکرار شد، بوسیله‌ی فرآیند ترکیب، اطلاعات بین memplex‌ها مبادله می‌شود. در طی فرآیند ترکیب، م‌ تحت تاثیر قورباغه‌هایی از memplex‌های مختلف قرار گرفته و کیفیت آن بهبود می‌یابد. بدین ترتیب فرآیند ترکیب تضمین می‌کند که تکامل فرهنگی به سوی هر هدفی که پیش می‌رود، بایاس نمی‌شود. جستجوی محلی و فرآیند ترکیب تا زمانی که معیار همگرایی معینی برآورده شود ادامه می‌یابند.

از یافته‌ها و تجربه‌های قبلی سایر اعضای گروه استفاده کنند. در مواقعی که منابع غذا به صورت غیرقابل پیش‌بینی در قسمت‌های مختلف توزیع شده‌اند، مزایای این روش بیشتر از معایب رقابت برای غذا است. نحوه‌ی حرکت منابع غذا غیرقابل پیش‌بینی است اما معمولاً مشابه حرکت افرادی است که مناسب‌ترین فاصله را با آنها دارند. پژوهشگران با مطالعه بر روی رفتارهای جوامع زیستی مشابه، دریافته‌اند که در جوامع زیستی مکانیزمی برای به اشتراک گذاشتن اطلاعات اجتماعی وجود دارد. این مکانیزم، امتیازی را برای تکامل جوامع زیستی و پایه‌ای را برای تشکیل SFL فراهم می‌کند.

الگوریتم SFL از تکامل ممتیک به شکل انتقال عقاید از یک فرد به سایر افراد در حین جستجوی محلی، استفاده می‌کند.

سپس با استفاده از استراتژی shuffling، اطلاعات بین جستجوهای محلی مبادله می‌شوند و الگوریتم به سوی بهینه‌ی سراسری حرکت می‌کند. درواقع، در این الگوریتم، مزایای MA‌های مبتنی بر ژنتیک و PSO مبتنی بر رفتار اجتماعی ترکیب می‌شوند. الگوریتم SFL اولین بار در سال ۲۰۰۳ توسط Eusuff و Lansey ایجاد شد [۳]. این الگوریتم مشابه PSO و GA یک الگوریتم بهینه‌سازی مبتنی بر جمعیت است که می‌توان از آن برای حل مسائل بهینه‌سازی پیچیده‌ای که غیرخطی^۷، تشخیص‌ناپذیر^۸ و چندوجهی^۹ هستند استفاده کرد. مهمترین مزیت SFL سرعت همگرایی بالای آن است. با این حال، در نسخه‌ی اصلی الگوریتم SFL، هر قورباغه موقعیت خود را بر اساس بهترین راه حل بهنگام (update) می‌کند و به دلیل اثری که بهترین راه حل محلی بر قورباغه‌ها می‌گذارد، تمام قورباغه‌ها به سرعت در اطراف آن جمع خواهند شد. در این مقاله به منظور بهبود پایداری این الگوریتم و همچنین برای افزایش توانایی آن برای جستجوی بهینه‌ی سراسری، یک مولفه‌ی شناختی به SFL اضافه می‌شود. با وجود این مولفه، هر قورباغه نه تنها موقعیت خود را بر اساس بهترین عضو memplex یا بر اساس بهترین عضو موجود در جمعیت تنظیم می‌کند، بلکه در این میان موقعیت خود را نیز می‌سنجد.

ادامه‌ی مقاله به این ترتیب سازماندهی شده است: در بخش بعدی شرح مختصری از نسخه‌ی اصلی الگوریتم SFL ارائه می‌شود. در بخش (۳) الگوریتم FSL بهبودیافته تشریح می‌شود و برای تسهیل پیاده‌سازی این الگوریتم فلوچارت آن ارائه می‌شود. نتایج تجربی در بخش (۴) ارائه و در بخش (۵) مورد بحث و بررسی قرار می‌گیرند. در نهایت بخش آخر شامل نتیجه‌گیری خواهد بود.

⁷ nonlinear

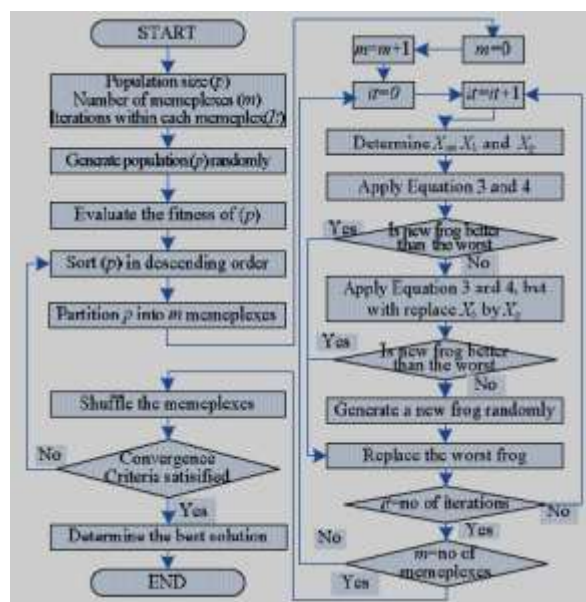
⁸ non-differentiable

⁹ multi-modal

تصادفی ایجاد و جایگزین بدترین قورباغه می‌شود. سپس این محاسبات به تعداد معینی تکرار می‌شوند.

۳. الگوریتم جهش ترکیبی قورباغی بهبودیافته

الگوریتم SFL مزایای MAهای مبتنی بر ژنتیک و الگوریتم‌های PSO مبتنی بر رفتار اجتماعی را ترکیب می‌کند و سعی می‌کند تا بین پیمایش عرضی فضای پاسخ بزرگ و جستجوی عمقی مکان‌های محتمل وجود بهینه‌ی سراسری، توازن برقرار کند. برای این منظور نیاز به یک فرآیند اجتماعی-روانشناختی بی‌هیچ گرایش خاصی به افراد است تا عقیده‌هایی که در گذشته برای افراد موفق نشان داده شده‌اند را منعکس کنند. یعنی افراد کارایی عقیده‌های اعضای همسایه-ی خود را مقایسه کنند و خود را نسبت به آنهایی تغییر دهند که نسبتاً موفق هستند.



شکل ۱- فلوچارت الگوریتم SFL بهبودیافته

شناخت را می‌توان به عنوان تفسیر قانون اثر Thorndike تلقی کرد، این قانون بیان می‌کند که اگر رفتاری که از روی تصادف صورت گرفته است تقویت شود، با احتمال بیشتری در آینده تکرار خواهد شد. هر فرد مختصات موقعیت کنونی خود را در فرمول وارد می‌کند و اینگونه مقدار اشتباهی که در تخمین مقادیر هدف مرتکب شده است را اندازه می‌گیرد، سپس به موقعیت جدیدی می‌رود و فرآیند را تکرار می‌کند. افراد در حین حرکت به سمت فضای چندمتغیره، مقدار

الگوریتم SFL ترکیبی از رویکردهای قطعی^{۱۰} و تصادفی^{۱۱} است. استراتژی قطعی به الگوریتم امکان می‌دهد تا از اطلاعات سطح پاسخ به صورت کارایی برای هدایت جستجوی هیوریستیک استفاده کند و مولفه‌های تصادفی، انعطاف-پذیری و استحکام جستجو را تضمین می‌کنند. SFL با جمعیت اولیه متشکل از "P" قورباغه آغاز می‌شود که به صورت تصادفی از فضای شدنی Ω ایجاد می‌شود. در مسائل D-بعدی، موقعیت آامین قورباغه به صورت $(x_{i1}, x_{i2}, \dots, x_{iD})$ نمایش داده می‌شود. سپس شایستگی هر قورباغه بر اساس موقعیتش محاسبه و قورباغه‌ها بر اساس میزان شایستگی شان به صورت نزولی مرتب می‌شوند. در مرحله‌ی بعدی کل جمعیت به m memplex تقسیم می‌شود. این تقسیم به نحوی انجام می‌شود که هر memplex شامل n قورباغه باشد (یعنی $P = m \times n$). در طی فرآیند تقسیم، اولین قورباغه در memplex اول قرار داده می‌شود، دومین قورباغه در memplex دوم، mامین قورباغه در memplex mام، قورباغه‌ی m+1 مجدداً در memplex اول و به همین ترتیب فرآیند ادامه می‌یابد.

در هر memplex، قورباغه‌هایی که بهترین و بدترین مقدار شایستگی را دارند به ترتیب با X_b و X_w نمایش داده می‌شوند. همچنین، قورباغه‌ای که بهترین میزان شایستگی را در کل جمعیت دارد با X_g نمایش داده می‌شود. سپس، با استفاده از یک فرآیند تکاملی، شایستگی بدترین قورباغه‌های موجود در هر چرخه از الگوریتم (نه همه‌ی قورباغه‌ها) اصلاح می‌شود. در طی فرآیند تکامل، هر قورباغه موقعیت خود را به نحوی اصلاح می‌کند که به بهترین قورباغه نزدیک شود. این اصلاح به صورت زیر انجام می‌شود:

(۱)

$$\text{Step size } S_i = \text{Rand}() \times (X_b - X_w)$$

(۲)

$$\text{New position } X_w = \text{Current position } X_w + S_i; \\ -S_{\max} \leq S_i \leq S_{\max}$$

که $\text{Rand}()$ یک عدد تصادفی در محدوده‌ی [0,1] است و S_{\max} حداکثر مقدار مجازی است که یک قورباغه بعد از اینکه تحت تاثیر قورباغه‌های دیگر قرار گرفت، می‌تواند جهش کند. چنانچه در طی این فرآیند راه‌حل بهتری تولید شود، راه‌حل بهتر جایگزین بدترین قورباغه خواهد شد. درغیراین‌صورت روابط (۱) و (۲) بر حسب بهترین قورباغه‌ی موجود در کل جمعیت، مجدداً محاسبه می‌شوند (یعنی X_g جایگزین X_b می‌شود و این روابط مجدداً محاسبه می‌شوند). اگر در این صورت هم هیچ بهبودی حاصل نشود، یک راه‌حل جدید به صورت

¹⁰ deterministic

¹¹ random

شوند. سپس دو بچمارک بهینه‌سازی گسسته مطرح می‌شوند. برای تسهیل ارزیابی، از زبان برنامه‌نویسی C++ برای پیاده‌سازی استفاده می‌شود.

۴.۱. چهار مساله‌ی بچمارک برای سنجش روش‌های

بهینه‌سازی پیوسته

مسائل بهینه‌سازی پیوسته ای که در این مقاله به عنوان بچمارک مورد استفاده قرار می‌گیرند، پیش از این در پژوهشات مختلف مطرح شده‌اند، بنابراین در ادامه فقط خلاصه‌ای از آنها ارائه می‌شود.

(۱) مساله‌ی آسان: این تابع هدف توسط Griewank مطرح شد و تابع هدفی مقیاس‌پذیر^{۱۲}، غیرخطی^{۱۳} و غیرقابل تفکیک^{۱۴} است که در آن پارامتر N (تعداد متغیرهای X_i) می‌تواند هر مقداری داشته باشد.

(۵)

$$f_1(x_i) = 1 + \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N (\cos(x_i / \sqrt{i})),$$

عبارت جمع در رابطه‌ی (۵)، شکلی سهمی‌وار دارد که تابع کسینوس موجود در عبارت ضرب، بر روی آن موج‌هایی را ایجاد می‌کند. این موج‌ها باعث ایجاد بهینه‌های محلی در فضای پاسخ می‌شوند. مقدار هر N متغیر X_i ، در محدوده -ی $[100, 100]$ قرار دارد. درواقعیکه تمام N متغیر برابر صفر هستند، بهینه‌ی سراسری تابع (مینیمم تابع) نیز برابر صفر خواهد بود.

(۲) مساله‌ی مسطح: مساله‌ی مسطح توسط Rosenbrock پیشنهاد شده است. این مساله پیش از این در [۴] مورد بحث و بررسی قرار گرفت. مینیمم این تابع، در نوک تپه‌ی همواری قرار دارد که تمییز آن از مکان‌های همسایه‌ی مشابه، دشوار و نیازمند جستجوی محلی دقیق می‌باشد. برخی از روش‌های بهینه‌سازی نمی‌توانند مینیمم این تابع را بیابند. بنابراین این مساله می‌تواند معیار خوبی برای سنجش قدرت روش‌های بهینه‌سازی باشد. فرمول این تابع برابر است با:

(۶)

$$f_2(x_i) = \sum_{i=1}^{N-1} [100 * (x_{i+1} - x_i^2) + (x_i - 1)^2]$$

این تابع به ازاء $x_i = 1$ و $i = 1, \dots, n$ یک مینیم $f_{min} = 0$ دارد. در ادامه شکل تعمیم‌یافته‌ی چندبعدی این تابع، پیاده‌سازی می‌شود.

اشتباه کنونی خود را با بهترین مقداری که تا آن زمان بدست آورده‌اند، مقایسه می‌کنند. بهترین ضریب خطایی که تا اینجا بدست آمده (که معادل با کمترین ضریب خطاست) $pbest_j$ نامیده می‌شود و مکانی که در آن این ضریب خطا بدست آمده به صورت بردار P_j نمایش داده می‌شود. خصیصه‌ای که این روش را از سایر روش‌های محاسبات تکاملی متمایز می‌کند، وابستگی آن به حافظه‌ی افراد برای به خاطر سپردن $pbest_j$ و P_j است. اگر موقعیت جدید از تمام موقعیت‌های پیشین بهتر باشد، آنگاه P_i با مختصات کنونی فرد بهنگام می‌شود. اختلاف بین P_i و X_i ، فاصله‌ی بین بهترین موقعیت قبلی و بهترین موقعیت کنونی را نشان می‌دهد. بردار اختلاف تمام افراد محاسبه می‌شود (بردار اختلاف، بردار حسرت گذشته "nostalgia vector" نام دارد). تمام مولفه‌های بردار اختلاف با یک عدد تصادفی مثبت در محدوده‌ی $[0, 1]$ وزن‌دهی می‌شوند. بردار جدید بدست‌آمده به صورت زیر با بردار تغییر جمع می‌شود:

$$S_i \leftarrow S_i + Rand() \times (P_i - X_i)$$

سپس با استفاده از روابط زیر، قورباغه‌ها دست‌کاری می‌شوند:

(۳)

$$\text{Step size } S_i = Rand() \times (P_w - X_w) + Rand() \times (X_b - X_w),$$

(۴)

$$\text{New position } X_w = \text{Current position } X_w + S_i, \\ -S_{max} \leq S_i \leq S_{max}.$$

با جمع بستن این بردار جدید با مختصات کنونی i (X_i) گرایشی برای بازگشت به سمت بهترین مکان قبلی فرد ایجاد می‌شود. از نظر روانشناختی این گرایش، تمایل برای یادآوری و بازگشت به بخش‌هایی از فضای روانی است که دلخواه فرد بوده‌اند، به عبارت دیگر رجوع به عقایدی که به نظر خوب یا تقویت‌کننده رسیده‌اند. در شکل (۱) فلوچارت الگوریتم SFL بهبودیافته ارائه شده است.

۴. ارزیابی تجربی ISFL

در این بخش میزان اثربخشی و کارایی الگوریتم ISFL با استفاده از بچمارک‌های مختلف ارزیابی می‌شود. برای این منظور در ادامه ابتدا چهار بچمارک بهینه‌سازی پیوسته که مقدار بهینه‌ی سراسری آنها مشخص است، معرفی می-

¹² scalable

¹³ non-linear

¹⁴ non-separable

(۱) مساله‌ی ۱:

$$f_5(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

With solution $x^* = (0, 0, 0, 0)^T$ and $f_5(x^*) = 0$.

(۲) مساله‌ی ۲:

$$f_6(x) = \|x\| = |x_1| + \dots + |x_D|,$$

که $x = (x_1, \dots, x_D) \in [-100, 100]^N$ است و N بعد مربوطه می‌باشد. در صورتیکه $f_6(x^*) = 0$ باشد، پاسخ برابر $x_i^* = 0, i = 1, \dots, N$ است.

۳.۴. نتایج تجربی

برای اینکه بتوان ISFL و SFL را مقایسه و میزان تاثیر مولفه‌ی شناختی بر روی SFL را بررسی کرد، در تمام آزمایشات از مقادیر یکسانی برای پارامترها استفاده شد. این مقادیر عبارتند از: اندازه‌ی جمعیت برابر ۲۰۰، تعداد memplex برابر ۲۰ و تعداد تکرار تکامل برابر ۱۰. دلیل انتخاب این مقادیر، گزارش ارائه شده در [۶] بود که حاکی از این است که در صورت بکارگیری این مقادیر، زمان پردازش و کیفیت پاسخ مطلوبی حاصل می‌شود. در کلیه‌ی آزمایشات، پروسه‌ی یافتن پاسخ در صورت برآورده شدن دو معیار زیر متوقف شد: (۱) دقت مورد نظر حاصل شد یا (۲) در طی g تکرار متوالی فرآیند ترکیب (shuffling)، بهبودی در تابع هدف ایجاد نشد.

(۱) نتایج بکارگیری SFL و ISFL بر روی مسائل بنچمارک

روش‌های بهینه‌سازی گسسته: در ازا هر مساله‌ی بهینه‌سازی و هر بنچمارک، ۱۰۰ اجرا انجام شد و کارایی متوسط روش بر حسب مقدار متوسط، نرخ موفقیت، متوسط تعداد تکرارها و بیشترین تعداد تکرار نمایش داده شد. تعداد تکرارها (g) به نوع مساله بستگی داشت. در مساله‌ی بنچمارک اول (تابع Griewank) با افزایش تعداد متغیرهای طراحی، تعداد مینیمم‌های محلی به صورت نمایی افزایش می‌یابد. نویسندگان این مقاله این مساله را با استفاده از ۱۰ متغیر طراحی، فرمول کردند؛ در این حالت تعداد تکرارها (g) به ۱۰۰ تا ۵۰۰ تکرار محدود می‌شود. در مساله‌ی بنچمارک دوم (تابع Rosen Brock) مقدار بهینه به ازا $x_i = 1, i = 1, \dots, n$ برابر $f_{opt} = 0$ است. نویسندگان این مقاله، مساله دوم را با استفاده از ۴ متغیر طراحی فرمول کردند؛ که در این حالت تعداد تکرارها (g) به ۱۰۰۰۰ تا ۳۰۰۰۰ محدود می‌شود. در مورد این مساله، تنها نتایج کمتر از ۰،۰۰۱، قابل قبول هستند. در فضای پاسخ مساله‌ی بنچمارک سوم (تابع Hubert)، ۱۸ مینیمم بهینه با مقدار

(۳) مساله‌ی چندوجهی با مقادیر مینیمم مشابه: تابع Shubert یک مساله‌ی چندوجهی است که قله‌های بهینه‌ی زیادی با مقادیر مشابه دارد. این تابع در فضای جستجو بیشتر از ۱۸ مینیمم بهینه دارد که مقدار همه‌ی آنها یکسان است. تابع Shubert، معیاری برای سنجش قابلیت روش‌های بهینه‌سازی است. درواقع روش بهینه‌سازی در صورتی روش قدرتمندی است که بتواند تمام مینیمم‌های این تابع را بیابد. فرمول این تابع عبارت است از:

(۷)

$$f_3(x_i) = \prod_{j=1}^n \left[\sum_{l=1}^5 j * \cos[(j+1) * x_l + j] \right],$$

$$j = 1, \dots, 5, i = 1, \dots, n, n = 2.$$

در ادامه شکل تعمیم‌یافته‌ی چندبعدی این تابع، پیاده‌سازی می‌شود.

(۴) مساله‌ی چندوجهی با نویز: این تابع که توسط Levy پیشنهاد شد، یک فضای جستجوی دوبعدی را تعریف می‌کند که تعداد زیادی مینیمم محلی دارد. این تابع شباهت زیادی با تابع Shubert دارد، اما در فرمول آن دو عدد بیشتر از فرمول تابع shubert وجود دارد که نویز و تغییر مقادیر مینیمم‌ها را مدل می‌کنند. این تابع معیاری برای سنجش قابلیت روش بهینه‌سازی در مدیریت محیط چندوجهی با نویز است. در [۵] از این تابع برای اجرای جستجو در الگوریتم PSO استفاده شده است. فرمول این تابع هدف عبارت است از:

(۸)

$$f_4(x_i) = \prod_{j=1}^n \left[\sum_{l=1}^5 j * \cos[(j+1) * x_l + j] \right],$$

$$+ (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2$$

$$j = 1, \dots, 5, i = 1, \dots, n, n = 2$$

۲.۴. دو مساله‌ی بنچمارک برای سنجش روش‌های بهینه‌سازی گسسته

در بسیاری از مسائل بهینه‌سازی، متغیرها نمی‌توانند مقادیر کسری داشته باشند و دادن مقادیر کسری به آنها کاری بی‌معنی است. در نتیجه بسیاری از این مسائل را می‌توان به صورت روش‌های بهینه‌سازی گسسته، مدل کرد. در این مقاله از دو مساله‌ی برنامه‌نویسی صحیح^{۱۵} به عنوان بنچمارک برای آزمایش کارایی ISFL استفاده شد. در این دو مسئله، بعد از اینکه موقعیت جدید هر قورباغه توسط رابطه (۴) مشخص می‌شود، قورباغه به نزدیکترین عدد صحیح تبدیل می‌شود. این دو مسئله‌ی بنچمارک که در ادامه تعریف می‌شوند، مکرراً در پژوهشات مرتبط بکار گرفته می‌شوند:

¹⁵ Integer programming

یکسانی 187.731- قرار دارد. در این مورد، روشی قابل قبول است که به دقت مطلوب ۰,۰۰۰۱ دست یابد. در مساله‌ی چهارم (تابع Levy) تابع هدف یک تابع آزمایش دوبعدی بدنام به نام Levy NO.5 است. در این مساله حدود ۷۶۰ بهینه‌ی محلی و یک بهینه‌ی سراسری وجود دارد و مقدار بهینه به ازاء $x_1 = -1.42513$ و $x_2 = -0.80032$ برابر $f_{opt} = -186.731$ است. در این مورد، روشی قابل قبول است که به دقت مطلوب ۰,۰۰۰۱ دست یابد. تعداد زیاد بهینه‌های محلی، یافتن بهینه‌ی سراسری را برای هر روشی مشکل می‌سازد.

نرخ موفقیت SFL و ISFL در حل تمام مساله‌های تست پیش گفته، بر حسب متوسط تعداد تکرار و بیشترین تعداد تکرار ثبت و در جدول (۱) نمایش داده شده است.

جدول ۱- تحلیل نتایج چهار مساله‌ی بهینه‌سازی پیوسته

test fun.	g	SFL			ISFL		
		Succ. ratio	Max. Its	Avg. Its	Succ. ratio	Max. Its	Avg. Its
f_1	100	47	94	84.15	94	79	42
	500	100	231	116.7	100	105	77.3
f_2	1000	38	953	891.6	65	958	955.38
	3000	100	2786	2161	99	2456	1873.6
f_3	50	85	47	35.29	99	39	31.18
	100	98	86	24.72	100	38	30.74
	1000	100	127	36.84	100	42	30.71
f_4	50	58	45	35.59	90	48	22.09
	100	84	85	41.72	93	92	26
	1000	98	973	82.95	99	310	30.95

جدول ۲- تحلیل نتایج دو مساله‌ی بهینه‌سازی گسسته

fun	g	SFL			ISFL		
		Succ. ratio	Avg. Its	Max. Its	Succ. ratio	Avg. Its	Max. Its
f_5	10	70	9.17	10	95	8.66	10
	50	100	9.63	40	100	10.44	24
	100	100	9.67	39	100	10.52	25
f_6	20	83	11.18	19	94	10.67	18
	50	85	15.4	47	93	10.93	49
	100	86	13.06	85	95	11.04	89

نتایج بکارگیری SFL و ISFL بر روی مسائل بنچمارک روش- های بهینه‌سازی گسسته: در این بخش نتایج کارایی الگوریتم‌های SFL و

ISFL بر روی دسته‌ای از مسائل که به عنوان مسائل برنامه‌نویسی صحیح معروفند، ارائه می‌شود. اینگونه که نتایج تجربی نشان می‌دهند تبدیل مقادیر حقیقی به اعداد صحیح تاثیر چندانی بر کارایی این دو الگوریتم نمی‌گذارد.

کارایی این دو الگوریتم در حل توابع f_5 و f_6 ، در طی ۱۰۰ آزمایش که تعداد تکرارهای متفاوت داشتند، آزمایش شد. نرخ موفقیت، متوسط تعداد تکرارها و حداکثر تعداد تکرارهای این دو الگوریتم در این ۱۰۰ آزمایش، در جدول (۲) ارائه شده است.

نتایج ارائه شده در جدول (۱) و (۲) حاکی از این هستند که SFL و ISFL قادرند هر ۵ مساله‌ی بنچمارک را مدیریت کنند، اما نتایج ISFL بهتر از نتایج SFL هستند.

۵. تحلیل نتایج تجربی

نتایج مربوط به مسائل بهینه‌سازی پیوسته که در جدول (۱) ارائه شده‌اند، نشان می‌دهند که ISFL می‌تواند هر ۴ مساله‌ی بنچمارک را حل کند و کارایی خوبی برای حل این مسائل دارد. در نسخه‌ی اصلی SFL، هر قورباغه موقعیت خود را بر اساس بهترین پاسخ بهنگام می‌کرد. در نتیجه به دلیل تاثیری که بهترین پاسخ محلی بر قورباغه‌ها می‌گذاشت، تمام قورباغه‌ها به سرعت در اطراف آن جمع می‌شدند. تعداد زیاد بهینه‌های محلی، یافتن بهینه‌ی سراسری را برای هر روشی مشکل می‌سازد. این قضیه در مساله‌ی بنچمارک ۴ که حدود ۷۶۰ بهینه‌ی محلی و یک بهینه‌ی سراسری دارد، تشدید می‌یابد. از این رو نرخ موفقیت SFL به ازاء ۵۰ تکرار، تنها برابر ۵۸٪ است. با اضافه کردن مولفه‌ی مشارکتی به این الگوریتم، نرخ موفقیت تا ۹۰٪ افزایش می‌یابد.

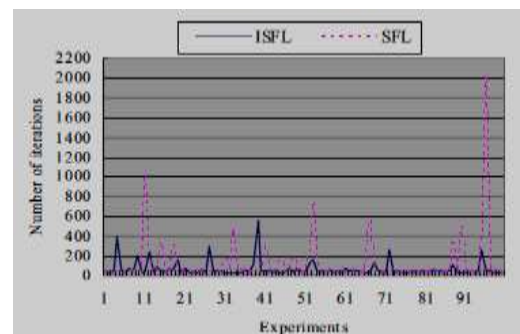
تعداد دفعاتی که ISFL می‌تواند مینیمم سراسری اولین مساله بهینه‌سازی گسسته (با ۱۰ تکرار) را بیابد، بیشتر از SFL است. این دو الگوریتم به ازاء تکرارهای ۵۰ و ۱۰۰، نرخ موفقیت مشابهی در حل مساله اول دارند، اما نرخ موفقیت ISFL در تعداد متوسط تکرارها بهتر از SFL است. نتایج اثبات می‌کنند که ISFL به خوبی روی مساله بنچمارک دوم عمل می‌کند.

به طور مختصر، نتایج ارائه شده در جدول (۱) و (۲) نشان می‌دهند که با اضافه کردن مولفه مشارکتی به SFL، گرچه متوسط تعداد تکرارها افزایش می‌یابد اما نرخ موفقیت نیز به مقدار زیادی افزایش می‌یابد. همگرایی سریع و قدرت الگوریتم برای رفع پیامدهای پیچیده، نیز از جمله مزایایی هستند که با افزایش پیچیدگی مساله آشکارتر می‌شوند.

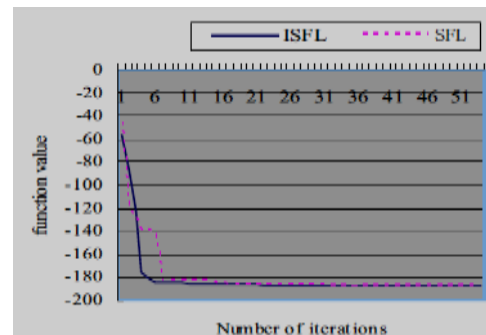
تابع f_4 تعداد زیادی مینیمم محلی دارد. الگوریتم ISFL خیلی زود به بهینه های محلی همگرا می شود اما همگرایی آن به بهینه ی سراسری با روند آهسته

تری صورت می گیرد. با این حال نتیجه ی نهایی این الگوریتم بهتر از الگوریتم های SFL است. درواقع اگر الگوریتم زمان کافی برای اجرا داشته باشد، به صورت بهتری می تواند از بهینه های محلی رهایی یابد.

نرخ های موفقیت بسیار بالای ISFL در شکل (۲) نشان از پایدار بودن این الگوریتم در ارتباط با تابع f_4 ، دارند. بعلاوه شکل (۳) نشان می دهد که این الگوریتم دچار ایستایی جستجو نمی شود. درواقع حرکت جمعی قورباغه ها به سمت بهترین موقعیت خودشان و حتی به سمت بهترین موقعیتی که توسط گروه کشف شده است و همچنین تغییر وزن موقعیت قبلی آنها، تضمین می کند که قورباغه ها دائما موقعیت خود را در حین بهینه سازی تغییر می دهند.



شکل ۲- تعداد تکرارها در ۱۰۰ اجرا



شکل ۳- همگرایی ISFL و SFL به بهترین مقدار تابع f_4

۶. نتیجه گیری

در تمام روش های بهینه سازی، تعیین مینیمم های سراسری کاری چالش برانگیز است. در این مقاله برای افزایش اثربخشی رویکردهای SFL، یک مولفه ی شناختی جدید ارائه می شود. مفروضات روانشناختی تئوری گروه قورباغه ها جامع و بی جدال است: یعنی قورباغه ها درحین جستجوی شناخت های

نامتناقض، بهترین عقیده های خود را نگه می دارند و از عقاید قورباغه های دیگر نیز استفاده می کنند. نتایج بکارگیری ISFL بر روی تعداد زیادی از بنچمارک های بهینه سازی دشوار، نشان می دهند که این الگوریتم دچار بهینه های محلی نمی شود و می تواند بهینه ی سراسری را بیابد. مولفه ی شناختی باعث پایداری همگرایی SFL می شود و بنابراین احتمال موفقیت این الگوریتم را افزایش می دهد. به نظر می رسد که ISFL حتی در مسائلی با ابعاد بالا نیز پایدار باشد. بعلاوه به نظر می رسد که این روش دچار ایستایی جستجو نمی شود. درواقع حرکت جمعی قورباغه ها به سمت بهترین موقعیت خودشان و حتی به سمت بهترین موقعیتی که توسط گروه کشف شده است و همچنین تغییر وزن موقعیت قبلی آنها، تضمین می کند که قورباغه ها دائما موقعیت خود را در حین بهینه سازی تغییر می دهند. کارهای آتی باید بر بهینه سازی کارایی الگوریتم ISFL پیشنهادی متمرکز شوند. بعلاوه بکارگیری ISFL برای حل مسائل بهینه سازی حقیقی پیچیده نیز جهت بررسی کامل ویژگی ها و ارزیابی کارایی مولفه ی شناختی، ضروری است.

۷. مراجع

- [1] Michalewicz. Z and Fogel. D, 2000, "How to Solve It: Modern Heuristics", Springer-Verlag, Berlin.
- [2] Kennedy. J and Eberhart. R, 1995, "Particle Swarm Optimization", in IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948, Perth.
- [3] Karaboga. D and Akay. B, 2009, "A Comparative Study of Artificial Bee Colony Algorithm", in Elsevier Journal on Applied Mathematics and Computation.
- [4] Eusuff. M and Lansey. K, 2003, "Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm", in Journal of Water Resource Plan and Management, Vol. 3, pp. 10-25.
- [5] Dorigo. M, Birattari. M, Stutzle. T, 2006, "Ant Colony Optimization", in IEEE Computational Intelligent Magazine, Vol. 1, pp. 28-39.
- [6] Amiri. B, Fathian. M and Maroosi. A, 2009, "Application of Shuffled Frog-Leaping Algorithm on Clustering", in journal of Springer on The International Journal of Advanced Manufacturing Technology, pp. 199-209.
- [7] Li. X, Lou. J, Chen. M. R and Wang. N, 2010, "An Improved Shuffled Frog-Leaping Algorithm with Extremal Optimisation for Continuous Optimisation", in journal of Elsevier on Information Sciences.