

تعیین موقعیت و تعقیب چندین بهینه در محیط‌های پویا با استفاده از الگوریتم

دسته ماهی‌های مصنوعی

بابک نصیری^۱؛ دانیال یزدانی^۲؛ محمد رضا میبدی^۳

چکیده

الگوریتم دسته ماهی‌های مصنوعی یکی از الگوریتم‌های هوش جمعی است که بر اساس جمعیت و جستجوی تصادفی کار می‌کند و کاربرد اصلی آن در حل مسائل بهینه‌سازی است. این الگوریتم در کاربردهای مختلفی از جمله خوشبندی داده‌ها، آموزش شبکه‌های عصبی، بهینه‌سازی توابع غیرخطی و غیره به کار رفته است. تا کنون تمام کاربردهای این الگوریتم، بهینه‌سازی در محیط‌های ایستا بوده است. در بسیاری از مسائل بهینه‌سازی در دنیای واقعی، تابع هدف یا محدودیت‌ها می‌توانند در طول زمان تغییر کنند که در نتیجه بهینه این مسائل نیز می‌توانند تغییر یابد. اگر هر یک از این رویدادهای نامعین در فرآیند بهینه‌سازی مورد توجه قرار گیرند، این مسئله پویا نامیده می‌شود. بسیاری از مسائل در دنیای واقعی به صورت پویا و غیرقطعی می‌باشند و حل آنها بصورت ایستا مناسب نیست. در این مقاله برای اولین بار یک الگوریتم دسته ماهی‌های مصنوعی برای بهینه‌سازی محیط‌های پویا ارائه می‌شود. نتایج حاصل از الگوریتم پیشنهادی بر روی بنچمارک قله‌های متحرک که در حال حاضر شناخته‌شده ترین معیار برای ارزیابی در محیط‌های پویا می‌باشد ارزیابی شده و با نتایج حاصل از چندین الگوریتم معتبر مورد مقایسه قرار گرفته است. نتایج بدست آمده نشان‌دهنده کارایی بالای الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها می‌باشد.

کلمات کلیدی

الگوریتم دسته ماهی‌های مصنوعی، محیط‌های پویا، چنددستگی، انحصار، معیار قله‌های متحرک.

A New Artificial Fish Swarm Algorithm for Dynamic Optimization Problems

Babak Nasiri; Danial Yazdani; MohammadReza Meybodi

Abstract

Artificial fish swarm algorithm is one of the swarm intelligence algorithms that works based on population and stochastic search and its main application is in solving optimization problems. This algorithm has been applied in such various applications as data clustering, neural networks learning, nonlinear function optimization and etc. Many problems in real world are dynamic and uncertain and it is not proper to solve them like solving static problems. In this paper, for the first time, an artificial fish swarm algorithm is presented for dynamic environments optimization. The results of the proposed approach were evaluated on moving peak benchmarks, which currently are the best known criterion for evaluation in dynamic environments, and were compared with results of multiple valid algorithms. Results show high efficiency of the proposed algorithm in comparison with other algorithms.

KEYWORDS

Artificial Fish Swarm Algorithm, Dynamic environments, Multi-swarm, Exclusion, Moving Peaks Benchmark.

^۱دانشگاه آزاد اسلامی قزوین، دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات، nasiri_babak@yahoo.com

^۲دانشگاه آزاد اسلامی شیراز، دانشکده مهندسی کامپیوتر، danial.yazdani@yahoo.com

^۳دانشگاه صنعتی امیرکبیر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، mmeybodi@aut.ac.ir

۱ - مقدمه

الگوریتم دسته ماهی‌های مصنوعی^۱ (AFSA) یکی از الگوریتم‌های برگرفته شده از طبیعت و هوش جمعی می‌باشد که توسط دکتر Li Xiao در سال ۲۰۰۲ ارائه شده است [۱]. این الگوریتم تکنیکی مبتنی بر رفتارهای جمعی است که از رفتارهای اجتماعی دسته ماهی‌ها در طبیعت الهام گرفته شده است. این الگوریتم دارای ویژگی‌های سرعت همگرایی بالا، غیرحساس بودن به مقادیر اولیه، انعطاف‌پذیری و تحمل‌پذیری خطاً بالایی است. از این الگوریتم در کاربردهای بهینه‌سازی از جمله یادگیری شبکه‌های عصبی پیشرو [۲]، خوشه‌بندی داده‌ها [۳][۴]، داده‌کاوی [۵]، بهینه‌سازی توابع غیرخطی [۶][۷]، بهینه‌سازی ترکیبی [۸][۹] و ... استفاده شده است.

عدم قطعیت در بسیاری از مسائل در دنیای واقعی کاملاً واضح و مشهود است. یکی از روش‌های برخورد با عدم قطعیت استفاده از روش‌های تکاملی و هوش جمعی می‌باشد. مسائل غیرقطعی که تاکنون توسط روش‌های تکاملی مورد بررسی قرار گرفته‌اند را بطور کلی می‌توان به چهار دسته تقسیم نمود. وجود نویز در تابع ارزیاب، آشفتگی در متغیرهای طراحی، تقریبی بودن تابع ارزیاب و پویا بودن راه حل بهینه. در این مقاله عدم قطعیت از نوع پویا بودن راه حل بهینه که جزو عمومی‌ترین انواع عدم قطعیت می‌باشد مورد توجه قرار گرفته است.

تاکنون روش‌های مختلفی برای حل مسائل پویا با استفاده از روش‌های پردازش تکاملی [۱۰] و هوش جمعی [۱۱] ارائه شده است. دو مشکل عمده روش‌های پردازش تکاملی که باعث عدم توانایی استفاده مستقیم از این روش‌ها برای بهینه‌سازی در محیط‌های پویا شده است، حافظه غیرمعتبر و از دست رفتن تنوع می‌باشد. هنگامی که محیط تغییر می‌کند، راه حل‌های بدست آمده موجود در حافظه دیگر معتبر نمی‌باشد و یا باید آنها بطور کامل فراموش کرد یا دوباره آنها را ارزیابی نمود. همچنین از آنجا که اکثر روش‌های پردازش تکاملی و هوش جمعی بدلیل ماهیتشان به یک نقطه همگرا می‌شوند لذا تنوع دسته در محیط از بین می‌رود و در صورت تغییر در محیط همگرا شدن به نقطه بهینه جدید در صورت امکان، بسیار زمانگیر خواهد بود. بنابراین می‌توان الزامات اولیه بهینه‌سازی در محیط‌های پویا را استفاده و به هنگام کردن صحیح حافظه و همچنین ایجاد تنوع در دسته‌ها پس از تغییر محیط دانست.

در این مقاله یک AFSA برای بهینه‌سازی در محیط‌های پویا ارائه می‌گردد که در آن تمام الزامات محیط‌های پویا برآورده شده است. در الگوریتم پیشنهادی مکانیزم رفتارهای پایه AFSA یعنی جستجوی غذا، دنباله‌روی، حرکت‌جمعی و حرکت‌آزادانه و همچنین روال اجرای AFSA [۱][۷][۹] تغییر کرده است. همچنین برخی از پارامترهای AFSA از جمله فاکتور شلوغی و طول گام حذف شده‌اند.

الگوریتم پیشنهادی بر روی بنچمارک دلهای متحرک (MPBⁱⁱ) [۱۲] که از معروف‌ترین بنچمارک‌های محیط‌های پویا است به کار رفته است و کارایی آن با شش الگوریتم دیگر به نامهای mCPSO [۱۳]، AmQSO [۱۴]، CellularPSCO [۱۵]، rSPSO [۱۶] مقایسه شده است. معیار مقایسه، خطای برونو خطیⁱⁱⁱ است که یکی از معیارهای اصلی مقایسه الگوریتم‌های طراحی شده برای محیط‌های پویا می‌باشد [۱۱]. نتایج آزمایشات نشان می‌دهد که الگوریتم پیشنهادی از کارایی قابل قبولی برخوردار است.

ادامه این مقاله بدین ترتیب سازماندهی شده است. بخش دوم الگوریتم پیشنهادی را تشریح می‌کند. در بخش سوم نتایج آزمایشات مورد بررسی قرار می‌گیرند و بخش نهایی به بیان نتیجه‌گیری می‌پردازد.

۲ - الگوریتم پیشنهادی

در این بخش یک الگوریتم جدید بر پایه الگوریتم دسته ماهی‌های مصنوعی برای بهینه‌سازی در محیط‌های پویا ارائه می‌گردد. در این الگوریتم رفتارها، پارامترها و روند اجرای AFSA تغییر کرده است تا الگوریتم پیشنهادی بتواند چندین قله را پیدا کند و آنها را پس از تغییر محیط دنبال کند. در الگوریتم پیشنهادی سه رفتار جستجوی غذا، دنباله‌روی و حرکت‌جمعی برای ماهی‌های مصنوعی انجام می‌شود که دارای تفاوت‌های عمدی‌ای با رفتارهای جستجوی غذا، دنباله‌روی و حرکت‌جمعی در AFSA استاندارد می‌باشد [۱][۷][۹]. در ادامه پس از تشریح الگوریتم دسته ماهی‌های مصنوعی جدید (NAFSA)، یک الگوریتم جدید بر اساس NAFSA برای محیط‌های پویا ارائه می‌شود.

۳ - الگوریتم دسته ماهی‌های مصنوعی جدید

ابتدا به شرح رفتارهای NAFSA می‌پردازیم.

• رفتار جستجوی غذا

این رفتار یک رفتار فردی است که هر ماهی مصنوعی بدون در نظر گرفتن دیگر اعضای دسته آن را اجرا می‌کند و به انجام یک جستجوی محلی در اطراف خود می‌پردازد. هر ماهی مصنوعی با اجرای این رفتار، به اندازه try_number بار تلاش می‌کند تا به یک موقعیت جدید با شایستگی بهتر نقل مکان کند. در اینجا فرض می‌شود ماهی مصنوعی آم در موقعیت \bar{X} قرار داشته و می‌خواهد رفتار جستجوی غذا را انجام دهد. در رفتار جستجوی غذا مراحل زیر انجام می‌شوند:

(۱) ماهی مصنوعی آم با استفاده از رابطه ۱ یک موقعیت هدف \bar{X}_T را در میدان دید خود در نظر می‌گیرد سپس مقدار شایستگی آنرا می‌سنجد. d نشان-دهنده شماره بعد می‌باشد و Rand یک عدد تصادفی را با توزیع یکنواخت در بازه [-۱,۱] تولید می‌کند.

$$X_{T,d} = X_{i,d} + Visual \times Rand_d(-1,1) \quad (1)$$

۲) در صورتی که مقدار شایستگی در موقعیت \vec{X}_T بهتر از موقعیت فعلی ماهی مصنوعی آم باشد، موقعیت ماهی مصنوعی آم با استفاده از رابطه ۲ بهنگام می‌شود.

$$\vec{X}_i = \vec{X}_T \quad (2)$$

مراحل ۱ و ۲ به میزان try_number بار اجرا می‌شوند.

با اجرای مراحل بالا، یک ماهی مصنوعی می‌تواند حداکثر try_number بار موقعیت خود را بهنگام کند و به سمت موقعیت‌های بهتر حرکت کند و این امکان نیز وجود دارد که هیچ یک از تلاش‌های آن برای یافتن موقعیت‌های بهتر نتیجه‌بخش نباشد و پس از اجرای رفتار جستجوی غذا هیچ تغییر مکانی ندهد.

• رفتار دنباله‌روی

همانطور که در بخش ۱-۱ شرح داده شد، در رفتار جستجوی غذا، هنگامی که یک ماهی مصنوعی موفق به جابه‌جایی به موقعیت‌های بهتر نشد، هیچ حرکتی نمی‌کند. در واقع برخلاف ماهی‌های مصنوعی در AFSA استاندارد که در صورت پیدا نکردن موقعیت‌های بهتر در رفتار جستجوی غذا، یک گام به صورت آزادانه و تصادفی حرکت می‌کنند [۱] و موقعیت قبلی خود را از دست می‌دهند، در NAFSA ماهی‌های مصنوعی در صورت عدم موقیت، موقعیت پیشین خود را حفظ می‌کنند. این امر موجب می‌شود که بهترین ماهی مصنوعی دسته در بهترین موقعیتی که تاکنون توسط اعضای دسته پیدا شده است قرار گیرد زیرا در رفتار جستجوی غذا در NAFSA. یک ماهی مصنوعی تنها در صورتی جابه‌جا می‌شود که به موقعیت بهتری نقل مکان کند. در رفتار دنباله‌روی، هر یک از ماهی‌های مصنوعی یک گام به سمت بهترین ماهی مصنوعی دسته با استفاده از رابطه ۳ حرکت می‌کنند:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \frac{\vec{X}_{Best} - \vec{X}_i(t)}{Dis_{i,Best}} \times [Visual \times Rand(0,1)] \quad (3)$$

که \vec{X} بردار موقعیت ماهی مصنوعی آم است که رفتار دنباله‌روی را اجرا می‌کند و \vec{X}_{Best} برابر با بردار موقعیت بهترین ماهی مصنوعی دسته است. بدین ترتیب ماهی مصنوعی آم حداکثر به اندازه Visual (میدان دید) در هر بعد به سمت بهترین ماهی مصنوعی دسته حرکت می‌کند. در واقع پس از اینکه یک ماهی غذای بیشتری را پیدا کند، دیگر اعضای دسته به دنبال آن حرکت می‌کنند تا آنها نیز به غذای بیشتر دست یابند. اجرای رفتار دنباله‌روی از بهترین ماهی مصنوعی دسته موجب افزایش سرعت همگرایی دسته می‌شود و به حفظ انسجام ماهی‌های مصنوعی در یک دسته کمک می‌کند. این رفتار، یک رفتار گروهی است و تعاملات بین اعضای دسته به صورت سراسری در میان کل اعضای دسته انجام می‌شود.

• رفتار حرکت جمعی

این رفتار نیز همانند رفتار دنباله‌روی یک رفتار گروهی است و به صورت سراسری در سطح اعضای دسته انجام می‌شود. در رفتار حرکت جمعی، ابتدا موقعیت مرکزی دسته که همان مرکز ثقل دسته است بر اساس میانگین حسابی موقعیت تمام اعضای یک دسته در هر بعد محاسبه می‌شود. برای ماهی مصنوعی آم شرط حرکت به سمت موقعیت مرکزی یعنی $f(X_{Center}) \geq f(X_i)$ چک می‌شود و در صورتی که این شرط برقرار بود، موقعیت بعدی ماهی مصنوعی آم با استفاده از رابطه ۴ بدست می‌آید:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \frac{\vec{X}_{Center} - \vec{X}_i(t)}{Dis_{i,Center}} \times [Visual \times Rand(0,1)] \quad (4)$$

رابطه ۴ برای حرکت به سمت موقعیت مرکزی تمام ماهی‌های مصنوعی دسته که موقعیتشان بدتر از موقعیت مرکزی است استفاده می‌شود اما برای بهترین ماهی مصنوعی دسته که در موقعیت \vec{X}_{Best} قرار دارد، در صورتی که مقدار شایستگی \vec{X}_{Center} بهتر از \vec{X}_{Best} باشد، موقعیت بعدی بهترین ماهی مصنوعی دسته از رابطه ۵ بدست می‌آید:

$$\vec{X}_{Best} = \vec{X}_{Center} \quad (5)$$

دلیل استفاده از رابطه ۵ برای بهترین ماهی مصنوعی دسته این است که ممکن است با حرکت به سمت موقعیت مرکزی با استفاده از رابطه ۴، در موقعیت بدتری از موقعیت فعلی اش قرار بگیرد. زیرا ممکن است در طول مسیر منتهی به سمت موقعیت مرکزی، موقعیت‌های بدتری وجود داشته باشد. در نتیجه این امر می‌تواند موجب از دست رفتن بهترین موقعیتی که توسط کل اعضای دسته تاکنون پیدا شده است شود که با استفاده از رابطه ۵ برای بهترین ماهی مصنوعی دسته این مشکل رفع می‌شود. دلیل استفاده نکردن از رابطه ۵ برای تمام ماهی‌های مصنوعی دسته این است که نقل مکان ماهی‌های دسته به یک موقعیت یکسان موجب کاهش شدید تنوع در گروه و پایین آمدن چشمگیر سرعت همگرایی می‌شود.

• روند اجرای NAFSA

در NAFSA برای هر یک از ماهی‌های مصنوعی در هر تکرار هر یک از رفتارهای جستجوی غذا، دنباله‌روی و حرکت جمعی انجام می‌شود. برخلاف AFSA استاندارد که اجرای یکی از دو رفتار حرکت جمعی و دنباله‌روی تأثیری در حرکت ماهی‌های مصنوعی نداشت و مقدار زیادی از محاسبات هدر می‌رفت، در

NAFSA هر سه رفتار جستجوی غذا، دنباله‌روی و حرکت‌جمعی در جایه‌جایی ماهی‌های مصنوعی و حرکت دسته به سمت موقعیت‌های بهتر تأثیرگذار هستند.

در NAFSA ابتدا تمامی ماهی‌های مصنوعی رفتار جستجوی غذا را اجرا می‌کنند و موقعیت آنها بر اساس روند اجرای رفتار جستجوی غذا بهنگام می‌شود. با اجرای این رفتار، هر یک از ماهی‌های مصنوعی می‌توانند تا try_number تغییر مکان را داشته باشند. سپس تمام آنها بر اساس موقعیت جدید خود و دیگر ماهی‌های مصنوعی دسته که رفتار جستجوی غذا را اجرا کرده‌اند، رفتار دنباله‌روی را انجام می‌دهند و تمام اعضا به غیر از بهترین ماهی مصنوعی دسته، به یک مکان جدید در راستای حرکت به سمت بهترین موقعیتی که توسط دسته پیدا شده است حرکت می‌کنند. در نهایت هر یک از ماهی‌های مصنوعی رفتار حرکت‌جمعی را انجام می‌دهند. با اجرای رفتار گروهی، ماهی‌های مصنوعی که دور از دسته مانده‌اند و در موقعیت‌های بدتری نسبت به دیگر اعضا دسته قرار گرفته‌اند، با سرعت بیشتری به دسته باز می‌گردند. اجرای این رفتار موجب افزایش سرعت همگرایی می‌شود و برخلاف رفتار دنباله‌روی حتی می‌تواند موجب بهبود موقعیت بهترین ماهی مصنوعی دسته شود.

۲ - پیکربندی NAFSA برای محیط‌های پویا

در MPB، هنگامی که بیش از یک قله در فضای مسئله وجود داشته باشد، هر قله می‌تواند پس از تغییر محیط تبدیل به بهینه سراسری شود، بنابراین باید تا جای ممکن تمام قله‌ها تحت پوشش ماهی‌های مصنوعی باشند تا اگر هر کدام از آنها تبدیل به بهینه سراسری شدند، الگوریتم بتواند به سرعت آنرا پیدا کند. بنابراین باید در هر قله یک دسته قرار گیرد و آنرا تحت پوشش قرار دهد. در این مقاله در ابتدای کار تها یک دسته در فضای مسئله وجود دارد، در صورتی که این دسته به یک قله همگرا شود یک دسته دیگر در محیط مسئله بوجود می‌آید. یک دسته هنگامی همگرا شده است که موقعیت بهترین ماهی دسته در فضای مسئله پس از چند تکرار تقریباً ثابت باشد. هنگامی که یک دسته همگرا شد مقدار پارامتر میدان دید آن کاهش می‌یابد تا دسته بتواند به خوبی جستجوی محلی را در نزدیکی هدف انجام دهد^[۹]. هر قله باید تنها تحت پوشش فقط یک دسته باشد زیرا افزایش تعداد دسته‌های موجود در یک قله تنها باعث مصرف بی‌دلیل تعداد ارزیابی شایستگی و در نتیجه افزایش سرعت تغییر محیط نسبت به تکرارهای اجرای الگوریتم می‌شود. بنابراین هنگامی که دو دسته بیش از یک حد معین به هم نزدیک شدند یعنی آنها به یک قله همگرا شده‌اند و در نتیجه دسته‌ای که مقدار شایستگی بهترین ماهی مصنوعی آن کمتر باشد از بین می‌رود. در الگوریتم پیشنهادی هرگاه تمام دسته‌های موجود در محیط مسئله همگرا شده بودند، یک دسته جدید در فضای مسئله مقداردهی اولیه می‌شود. در هر تکرار تنها یک دسته آزاد باید وجود داشته باشد. دسته آزاد دسته‌ای است که هنوز همگرا شده باشد. هنگامی که بیش از یک دسته آزاد باشد، تنها بهترین آنها در فضای مسئله باقی می‌ماند و بقیه آنها از بین می‌روند. در نتیجه پس گذشت مدتی از اجرای مسئله تمام قله‌ها تحت پوشش قرار می‌گیرند و تنها یک دسته آزاد در فضای مسئله وجود خواهد داشت.

برای کشف تغییر در محیط یک نقطه تصادفی در اجرای الگوریتم در نظر گرفته می‌شود و مقدار شایستگی آن ثبت می‌شود. در هر تکرار از اجرای الگوریتم مقدار شایستگی این نقطه ارزیابی می‌شود و در صورتی که با مقدار قبلی آن متفاوت باشد یعنی محیط تغییر کرده است. پس از کشف تغییر در محیط ابتدای مقدار شایستگی تمام ماهی‌های مصنوعی در تمام دسته‌ها ارزیابی می‌شود تا مقدار شایستگی ثبت شده برای هر ماهی مصنوعی معتبر باشد و الگوریتم بتواند پس از آن به درستی رفتارهایش را اجرا کند. سپس در هر دسته تنها بهترین ماهی مصنوعی دسته در جای خود باقی می‌ماند و دیگر اعضا در محدوده یک شاعع خاص در اطراف آن به صورت تصادفی پخش می‌شوند. مقدار این شاعع برابر با طول جایه‌جایی قله‌ها در نظر گرفته می‌شود. بدین ترتیب با احتمال زیاد پس از تغییر نیز نوک قله (هدف) در محدوده دید اعضا دسته قرار دارد. همچنین با این عمل تنوع در دسته افزایش می‌یابد.

۳ نتایج آزمایشات

برای ارزیابی صحت و کارایی الگوریتم پیشنهادی، این الگوریتم همراه با شش الگوریتم شناخته شده به به نام‌های mCPSO^[۱۳]، AmQSO^[۱۴]، CellularPSO^[۱۵]، SPSO^[۱۶] و rSPSO^[۱۷] بر روی MPB مورد مقایسه قرار گرفته‌اند. همچنین دو الگوریتم mCPSO و mQSO همراه با ضدهمگرایی نیز در مقایسه اورده شده‌اند. نتایج با توجه به سناریو ۲ در مسئله MPB بدست آمده‌اند^[۱۲]. تنها پارامتر متفاوت، تعداد قله‌ها می‌باشد که برای ارزیابی بهترین روش‌ها از ۱ قله تا ۲۰۰ قله در فرکانس تغییر ۵۰۰۰ در نظر گرفته شده است. در آزمایشات در الگوریتم پیشنهادی تعداد ماهی‌های مصنوعی در هر دسته برابر ۲ و مقدار Try_number برابر در نظر گرفته شده. مقدار Visual Try_number برای دسته‌هایی که همگرا نشده‌اند برابر ۲۰ و برای دسته‌هایی که همگرا شده‌اند برابر ۵. در نظر گرفته می‌شوند. مقدار این پارامترها بر اساس آزمایشات بسیار زیاد تعیین شده است.

آزمایشات ۳۰ بار انجام شده‌اند و نتایج بدست آمده از آزمایشات (خطای برون خطی و خطای استاندارد)^۴ بر روی MPB با ۱۰۰ بار تغییر محیط در هر آزمایش بر روی تعداد قله‌های مختلف با فرکانس تغییرات محیط ۵۰۰۰ در جدول (۲) نشان داده شده است. خطای استاندارد در کنار خطای برون خطی درون پرانتر نشان داده شده است. دو نتیجه بهتر در هر سطر پررنگ شده است و بهترین نتیجه به صورت کج می‌باشد. در این جدول ها P برابر تعداد قله‌ها می‌باشد. دو الگوریتم mCPSO و mQSO همراه با ضدهمگرایی با ستاره مشخص شده‌اند.

همانطور که در جدول (۲) مشاهده می‌شود، الگوریتم پیشنهادی در تمام موارد به نتایج بهتری نسبت به دیگر الگوریتم‌ها دست یافته است. از آنجایی که تعداد دسته‌ها در الگوریتم پیشنهادی بر اساس تعداد قله‌ها تعیین می‌شود، بنابراین کارایی الگوریتم در تعداد قله‌های پایین بسیار قابل قبول است. این برتری

در AmQSO نیز وجود دارد زیرا در این الگوریتم نیز تعداد دسته‌ها منطبق بر تعداد قله‌های یافتشده تعیین می‌شود اما در الگوریتم‌های دیگر به دلیل ثابت بودن تعداد دسته‌ها، در مواقعي که تعداد دسته‌ها بیشتر از تعداد قله‌ها باشد دسته‌ها بر سر قله‌ها با یکدیگر به رقابت می‌پردازن و با استفاده از انحصار یکدیگر را از قله‌ها بیرون می‌کنند، که در نتیجه آن نتایج بدست آمده، کمتر از حد انتظار برای تعداد قله‌های پایین می‌باشد.

برای الگوریتم پیشنهادی بالا رفتن تعداد قله‌ها، کارایی الگوریتم پایین می‌آید. این مسئله از آنجا ناشی می‌شود که هنگامی که تعداد قله‌ها زیاد باشد، دسته‌های بیشتری در فضای مسئله وجود خواهد داشت. با افزایش تعداد دسته‌ها، تعداد ارزیابی شایستگی در هر تکرار بالا می‌رود و از آنجایی که فرانکس تغییر محیط بر اساس تعداد ارزیابی شایستگی است، باعث می‌شود که ماهی‌های مصنوعی تعداد تکرار کمتری را تغییر محیط بعدی پیش رو داشته باشند. در نتیجه دقت جستجو پایین‌تر می‌آید و نتایج با کاهش کیفیت مواجه می‌شوند. الگوریتم AmQSO بیش از الگوریتم پیشنهادی با این مشکل مواجه است و در تعداد قله‌های ۱۰۰ و ۲۰۰ کارایی آن حتی از مدل قبلی آن یعنی mQSO که در آن تعداد دسته‌ها ثابت است نیز کمتر شده است. در الگوریتم پیشنهادی بدلیل افزایش سریع تنوع پس از تغییر محیط و پوشش دادن سریع نوک قله، نتایج با بالا رفتن تعداد قله‌ها کمتر با کاهش کیفیت مواجه می‌شوند.

۴ نتیجه‌گیری

در این مقاله برای اولین بار یک روش نوین برای بهینه‌سازی در محیط‌های پویا بر اساس الگوریتم دسته ماهی‌های مصنوعی پیشنهاد شد و نتایج آن بر روی تابع بنچمارک قله‌های متحرک چندین روش شناخته شده دیگر مورد مقایسه قرار گرفت. نتایج آزمایشات نشان داد که الگوریتم پیشنهادی از کارایی قابل قبولی در تمامی موارد یعنی هم در تعداد قله‌های کم و هم در تعداد قله‌های زیاد برخوردار است.

جدول(۲) : خطای برون خطی و خطای استاندارد الگوریتم‌ها در فرانکس تغییرات ۵۰۰۰ با تعداد قله‌های مختلف.

P	Cellular PSO	mCPSO*	mQSO*	mCPSO	mQSO	AmQSO	rSPSO	SPSO	الگوریتم پیشنهادی
۱	۲.۵۵(۰,۱۲)	۴,۹۳(۰,۱۷)	۵,۰۷(۰,۱۷)	۴,۹۳(۰,۱۷)	۵,۰۷(۰,۱۷)	۰,۶۱(۰,۱۲)	۱,۴۲(۰,۰۶)	۲,۶۴(۰,۱۰)	۰,۵۹(۰,۰۶)
۵	۱,۶۸(۰,۱۱)	۲,۰۷(۰,۰۸)	۱,۸۱(۰,۰۷)	۲,۰۷(۰,۰۸)	۱,۸۱(۰,۰۷)	۱,۰۱(۰,۰۹)	۱,۰۴(۰,۰۳)	۲,۱۵(۰,۰۷)	۰,۶۶(۰,۰۵)
۱۰	۱,۷۸(۰,۰۵)	۲,۰۸(۰,۰۷)	۱,۸۰(۰,۰۶)	۲,۰۵(۰,۰۷)	۱,۷۵(۰,۰۶)	۱,۵۱(۰,۱۰)	۱,۵۰(۰,۰۸)	۲,۵۱(۰,۰۹)	۰,۹۵(۰,۰۴)
۲۰	۲,۶۱(۰,۰۷)	۲,۶۴(۰,۰۷)	۲,۴۲(۰,۰۷)	۲,۹۵(۰,۰۸)	۲,۷۴(۰,۰۷)	۲,۰۰(۰,۰۱۵)	۲,۲۰(۰,۰۷)	۳,۲۱(۰,۰۷)	۱,۳۹(۰,۰۵)
۳۰	۲,۹۳(۰,۰۸)	۲,۶۳(۰,۰۸)	۲,۴۸(۰,۰۷)	۳,۳۸(۰,۱۱)	۳,۲۷(۰,۱۱)	۲,۱۹(۰,۱۷)	۲,۶۲(۰,۰۷)	۳,۶۴(۰,۰۷)	۱,۶۰(۰,۰۶)
۴۰	۳,۱۴(۰,۰۸)	۲,۶۷(۰,۰۷)	۲,۵۵(۰,۰۷)	۳,۶۹(۰,۱۱)	۳,۶۰(۰,۰۸)	۲,۲۸(۰,۱۲)	۲,۷۶(۰,۰۸)	۳,۸۵(۰,۰۸)	۱,۶۰(۰,۰۵)
۵۰	۳,۲۶(۰,۰۸)	۲,۶۵(۰,۰۶)	۲,۵۰(۰,۰۶)	۳,۶۸(۰,۱۱)	۳,۶۵(۰,۱۱)	۲,۴۳(۰,۱۳)	۲,۷۲(۰,۰۸)	۳,۸۶(۰,۰۸)	۱,۱۱(۰,۰۶)
۱۰۰	۳,۴۱(۰,۰۷)	۲,۴۹(۰,۰۴)	۲,۳۶(۰,۰۴)	۴,۰۷(۰,۰۹)	۳,۹۳(۰,۰۸)	۲,۶۸(۰,۱۲)	۲,۹۳(۰,۰۶)	۴,۰۱(۰,۰۷)	۱,۹۲(۰,۰۵)
۲۰۰	۳,۴۰(۰,۰۶)	۲,۴۴(۰,۰۴)	۲,۲۶(۰,۰۳)	۳,۹۷(۰,۰۸)	۳,۸۶(۰,۰۷)	۲,۶۲(۰,۱۰)	۲,۷۹(۰,۰۵)	۳,۸۲(۰,۰۵)	۱,۹۷(۰,۰۵)

۵ مراجع

- [۱] L. X. Li, Z. J. Shao and J. X. Qian, “An Optimizing Method based on Autonomous Animals: Fish Swarm Algorithm”, In Proc. of Systems Engineering Theory & Practice, ۲۰۰۲.
- [۲] C. R. Wang, C. L. Zhou and J. W. Ma, “An Improved Artificial Fish Swarm Algorithm and Its Application in Feed-Forward Neural Networks”, In Proc. of ۴th International Conference on Machine Learning and Cybernetics, Guangzhou, ۲۰۰۵.
- [۳] D. Yazdani, B. Saman and M. R. Meybodi, “A Modified Artificial Fish Swarm Algorithm for Data Clustering”, in ۴th National Conference on Commands and Control(C&I), Tehran, Iran. ۲۰۱۰.
- [۴] D. Yazdani, S. Golyari and M. R. Meybodi, “A New Hybrid Approach for Data Clustering”, in ۵th International Conference on Telecommunication (IST۲۰۱۰), Tehran, Iran, ۲۰۱۰.
- [۵] M. Zhang, C. Shao, M. Li and J. Sun, “Mining Classification Rule with Artificial Fish Swarm”, in ۶th World Congress on Intelligent Control and Automation, Vol. ۲, pp. ۵۸۷۷-۵۸۸۱, Dalian, ۲۰۰۶.
- [۶] D. Yazdani and A. T. Haghigiat, “A Modified Artificial Fish Swarm Algorithm with New Behavior for Avoiding Trap in Local Optima”, in ۱۷th Iranian Conference on Electrical Engineering(ICEE۲۰۰۹), Tehran, Iran, ۲۰۰۹.
- [۷] D. Yazdani, Adel Nadjaran and M. R. Meybodi, “Fuzzy Adaptive Artificial Fish Swarm Algorithm”, in ۲۲th international conference on artificial intelligence, Adelaide, Australia, ۲۰۱۰.

- [۸] D. Yazdani, S. Golyari and M. R. Meybodi, "A New Hybrid Algorithm for Optimization Based on Artificial Fish Swarm Algorithm and Cellular Learning Automata", in ۵th International Conference on Telecommunication (IST۲۰۱۰), Tehran, Iran, ۲۰۱۰.
- [۹] D. Yazdani and M. R. Meybodi, "AFSA-LA: A New Model for Optimization", in ۱۵th Conference of Computer Society of Iran (CSICC۲۰۱۰), Tehran, Iran, ۲۰۱۰.
- [۱۰] Y. Jin and J. Branke, "Evolutionary Optimization in uncertain environments –A Survey", in IEEE Transaction on Evolutionary Computation, vol. ۹, No. ۳, pp. ۳۰۳-۳۱۷, ۲۰۰۵.
- [۱۱] S. Yang and C. Li, "A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments", in IEEE Transaction on Evolutionary Computation, pp. ۱-۱۶, ۲۰۱۰.
- [۱۲] <http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/>
- [۱۳] T. Blackwell and J. Branke, "Multiswarm, Exclusion, and Anti-Convergence in Dynamic Environment", in IEEE Transaction on Evolutionary Computation, Vol. ۱۰, No. ۴, pp. ۴۵۹-۴۷۲, ۲۰۰۶.
- [۱۴] T. Blackwell and J. Branke, "Particle Swarms for Dynamic Optimization Problems", in Swarm Intelligence, pp. ۱۹۳-۲۱۷, ۲۰۰۸.
- [۱۵] B. Hashemi and M. R. Meybodi, "Cellular PSO: A PSO for Dynamic Environments", in Advances in Computation and Intelligence, Lecture Notes in Computer Science, vol. ۵۸۲۱, pp. ۴۲۲-۴۳۳, ۲۰۰۹.
- [۱۶] W. Du and B. Li, "Multi-Strategy Ensemble Particle Swarm Optimization for Dynamic Optimization", in Information Sciences: an International Journal Vol. ۱۷۸, pp. ۳۰۹۶-۳۱۰۹, ۲۰۰۸.
- [۱۷] D. Parrott and X. Li, "Locating and Tracking Multiple Dynamic Optima by A Particle Swarm Model Using Speciation," in IEEE Transaction on Evolutionary Computation, vol. ۱۰, No. ۴, pp. ۴۴۰-۴۵۸, ۲۰۰۶.
- [۱۸] S. Bird and X. Li, "Using regression to improve local convergence," in Proc. IEEE Congress Evolutionary Computation, ۲۰۰۷, pp. ۵۹۲-۵۹۹.

زیرنویس‌ها

ⁱ Artificial Fish Swarm Algorithm

ⁱⁱ Moving Peak Benchmark

ⁱⁱⁱ Offline Error

^{iv} Standard Error

کنفرانس داده کاوی ایران