

AFSA-LA: یک مدل جدید برای بهینه‌سازی

محمد رضا میدی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی

امیرکبیر، تهران، ایران

mmeybodi@aut.ac.ir

دانیال یزدانی

دانشکده برق، رایانه و فناوری اطلاعات دانشگاه آزاد اسلامی قزوین،

ایران

d_yazdani@qiau.ac.ir

چکیده- الگوریتم دسته ماهی‌های مصنوعی (AFSA) یک مدل محاسباتی تکاملی است که مبتنی بر الگوریتم‌های هوش جمعی می‌باشد. عملکرد این الگوریتم بر اساس جستجوی تصادفی است و نمونه‌ای از رفتارگرایی در هوش مصنوعی می‌باشد. در این روش ماهی‌های مصنوعی با اجرای رفتارهایی که از طبیعت برگرفته شده‌اند به سمت بهینه سراسری حرکت می‌کنند. مشکل اصلی این روش این است که با تنظیم پارامترهای آن، الگوریتم تنها قادر است یکی از جستجوهای محلی یا سراسری را به خوبی انجام دهد. در این مقاله یک مدل جدید بر اساس AFSA به نام AFSA-LA پیشنهاد می‌گردد که در آن از اتماتای یادگیر برای تنظیم رفتارهای ماهی‌های مصنوعی و کنترل پارامترهای آنها استفاده می‌شود که در آن بین جستجوی سراسری و جستجوی محلی موازن برقرار شده است. نتایج آزمایشات بر روی مسائل نمونه نشان می‌دهند که روش پیشنهادی دارای عملکرد بهتری در مقایسه با مدل AFSA استاندارد است.

واژه‌های کلیدی- الگوریتم دسته ماهی‌های مصنوعی، اتماتای یادگیر، بهینه‌سازی.

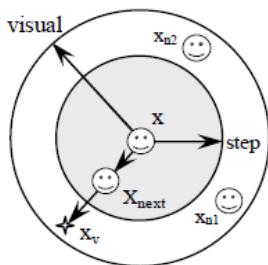
الگوریتم‌ها در حل مسائل بهینه‌سازی می‌باشد. الگوریتم دسته ماهی-

های مصنوعی^۱ AFSA گونه‌ای از الگوریتم‌های برگرفته شده از طبیعت و الگوریتم‌های هوش جمعی می‌باشد که توسط دکتر Li Xiao Lei در سال ۲۰۰۲ ارائه شده است [۲]. این الگوریتم یک تکنیک جدید مبتنی بر رفتارهای جمعی است که از رفتارهای اجتماعی دسته ماهی‌ها در طبیعت الهام گرفته شده است. این الگوریتم دارای ویژگی‌های سرعت طبیعیت بالا، غیرحساس بودن به مقادیر اولیه، انعطاف‌پذیری و تحمل-پذیری خطای بالایی است. این الگوریتم در کاربردهای بهینه‌سازی از جمله تخمین پارامترها [۳،۴]، بهینه‌سازی ترکیبی [۵]، یادگیری شبکه‌های عصبی پیشرو [۶] و ... به کار رفته است. در AFSA دو پارامتر مسافتی به نام‌های میدان دید و طول گام وجود دارد. ماهی‌های مصنوعی در فضای میدان دید خود به جستجوی فضای مسئله می-پردازند و به اندازه درصدی تصادفی از طول گام به سمت هدف خود در هر تکرار حرکت می‌کنند. در AFSA استاندارد، تعیین مقدار اولیه این دو پارامتر تأثیر بسزایی در کیفیت نتیجه نهایی دارد. مقدار این دو پارامتر تا انتهای اجرای الگوریتم ثابت و برابر با همان مقدار اولیه آنها باقی می‌ماند. در صورتی که مقدار اولیه این دو پارامتر را بزرگ در نظر بگیریم دسته ماهی‌های مصنوعی می‌توانند با سرعت بیشتری به سمت بهینه سراسری حرکت کنند و توانایی بیشتری برای عبور از بهینه‌های محلی خواهند داشت و در صورتی که مقدار این دو پارامتر را کوچک در نظر بگیریم الگوریتم می‌تواند با دقیق بیشتری جستجوی محلی را انجام دهد. بنابراین با تعیین مناسب مقدار اولیه این پارامترها، الگوریتم می‌تواند یکی از جستجوهای محلی و یا جستجوی سراسری را به طور مطلوب انجام دهد ولی نمی‌تواند هر دوی آنها را در یک اجرا به خوبی انجام دهد.

۱- مقدمه

مفهوم بهینه‌سازی بدین صورت است که در بین پارامترهای یک تابع به دنبال مقادیری باشیم که تابع را کمینه یا بیشینه می‌نمایند. کلیه مقادیر مناسب جهت این امر را، راه حل‌های ممکن و بهترین مقدار از این مقادیر را راه حل بهینه می‌نامند. الگوریتم‌های بهینه‌سازی هر دو نوع مسائل بیشینه‌سازی و کمینه‌سازی را پوشش می‌دهند. بهینه‌سازی کاربردهای زیادی دارد از جمله در تخصیص منابع، زمان‌بندی‌ها، تصمیم‌گیری‌ها و روش‌های مختلفی برای بهینه‌سازی وجود دارد [۱]. یافتن راه حل برای مسائلی که از نوع NP می‌باشند بسیار مشکل است. الگوریتم‌هایی از جمله الگوریتم‌های هوش جمعی تا حدی این مشکل را حل نموده‌اند. توسط این نوع الگوریتم‌ها راه حل‌هایی پیدا می‌شوند که تقریباً به جواب نزدیکند. هوش جمعی نوعی روش هوش مصنوعی است که مبتنی بر رفتارهای جمعی در سامانه‌های نامتمرکز و خودسامانده بنيان شده است. این سامانه‌ها معمولاً از جمعیتی از عامل‌های ساده تشکیل شده‌اند که بطور محلی با یکدیگر و با محیط خود در تعامل هستند. با وجود اینکه معمولاً هیچ کنترل تمرکزیافته‌ای، چگونگی رفتار عامل‌ها را به آنها تحمیل نمی‌کند، تعاملات محلی آنها به پیدایش رفتاری عمومی می‌انجامد. موفق ترین روش‌های هوش مصنوعی جمعی که تاکنون بوجود آمده‌اند، روش بهینه‌سازی گروه مورچه‌ها، روش بهینه‌سازی دسته ذرات و روش بهینه‌سازی گروه ماهی‌ها هستند [۱]. کاربرد این الگوریتم‌ها در انواع زمینه‌های تحقیقات علمی نشان‌دهنده عملی بودن و ویژگی‌های مطلوب آنها است. از جمله ویژگی‌های این الگوریتم‌ها می‌توان به مقایسه‌پذیری، تحمل‌پذیری خطای، سازگاری، سرعت، استقلال داخلی، موازنی بودن و ... اشاره کرد. کاربرد عمده این

محیطی که ماهی مصنوعی در آن زندگی می‌کند اساساً فضای راه حل و حوزه‌های ماهی‌های مصنوعی دیگر است. درجه تراکم غذا در منطقه آبی تابع هدف AFSA می‌باشد. در طول فرآیند بهینه‌سازی AFSA، اطلاعات فردی و محیطی، کاملاً جهت جستجو تا رسیدن به توازن استفاده می‌شوند. در نهایت، ماهی‌های مصنوعی به مکانی می‌رسند که درجه تراکم و غلظت غذا در آنجا بیشترین (بهینه سراسری) باشد. همان‌طور که در شکل ۱ مشاهده می‌شود، ماهی مصنوعی مفاهیم خارجی را از طریق بینایی درک می‌کند [۸].



شکل ۱: ماهی مصنوعی و محیط پیرامون آن.

وضعیت فعلی ماهی مصنوعی توسط بردار $X = (x_1, x_2, \dots, x_n)$ نشان داده می‌شود که ($i=1, 2, \dots, n$) متغیری است که برای مقدار بهینه (متغیر بهینه‌سازی) جستجو می‌شود. Visual برابر میدان دید ماهی مصنوعی می‌باشد و X_v موقعیتی در میدان دید است که ماهی مصنوعی می‌خواهد به آنجا برود. حال اگر وضعیت $(x_1^v, x_2^v, \dots, x_n^v)$ از X_v از نظر تراکم غذایی بهتر از وضعیت فعلی باشد، یک گام به جلو در جهت آن پیش می‌رویم که باعث تغییر وضعیت ماهی مصنوعی از X به X_{next} می‌شود، ولی اگر وضعیت فعلی بهتر از X_v باشد، به گشتزنی در محدوده میدان دید ادامه می‌دهیم. گشتزنی‌های بیشتر ماهی مصنوعی باعث ایجاد داشش و اطلاعات درباره کل اوضاع میدان دید می‌شود. Step برابر حداکثر طول گام حرکت است، فاصله بین دو ماهی مصنوعی که در وضعیت‌های X_i و X_j هستند با $\|X_i - X_j\| = d$ نشان داده می‌شود. در رفتارهای ماهی مصنوعی از یکتابع به نام تابع Rand() استفاده می‌شود. تابع Rand() تابعی است که یک عدد تصادفی با توزیع یکنواخت بین صفر و یک تولید می‌کند.

مدل ماهی مصنوعی شامل دو بخش متغیرها و توابع است که متغیرها شامل X (موقعیت فعلی ماهی مصنوعی)، Step (حداکثر طول گام حرکت)، Visual (میدان دید)، try_number (سقف تعداد آزمایش و سعی کردن) و فاکتور شلوغی δ ($0 < \delta < 1$) می‌باشد [۹]. همچنین توابع شامل رفتار جستجوی (شکار) غذا، رفتار حرکت گروهی و رفتار دنباله‌روی (پیروی) می‌باشند. نحوه اجرای این توابع در هر تکرار الگوریتم به این صورت است که در هر تکرار، دو رفتار حرکت گروهی و دنباله‌روی اجرا می‌شوند و در صورتی که هر کدام از این دو رفتار به نتیجه مورد نظر نرسند، رفتار جستجوی غذا در ادامه آنها اجرا می‌گردد.

۲-۱- رفتار جستجوی غذا

یک اتوماتای یادگیر [۱۰] ماشینی است که می‌تواند تعدادی متناظر عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتای یادگیر داده می‌شود و اتوماتای یادگیر از این پاسخ در انتخاب عمل بعدی استفاده می‌کند و بدین ترتیب به سمت انتخاب عملی که بیشترین پاداش را از محیط می‌گیرد میل می‌کند. به عبارت بهتر، اتوماتا عملی را که بیشترین پاداش را از محیط دریافت می‌کند یاد می‌گیرد. اتوماتای یادگیر برای بهبود قدرت یادگیری بسیاری از الگوریتم‌ها مورد استفاده قرار گرفته است که از آن جمله می‌توان به شبکه‌های عصبی [۱۱]، الگوریتم‌های ژنتیک [۱۲] و حرکت دسته جمعی ذرات [۱۳] اشاره نمود.

در این مقاله یک مدل جدید به نام AFSA-LA پیشنهاد می‌گردد. در مدل پیشنهادی، از دو اتوماتای یادگیر مختلف برای کنترل پارامترهای ماهی‌های مصنوعی و تنظیم رفتارهای آنها استفاده شده است. اتوماتای یادگیر با کنترل اندازه پارامترهای میدان دید و طول گام در طول تکرارهای اجرای الگوریتم نوعی تعادل میان جستجوی محلی و جستجوی سراسری ایجاد می‌کند. در واقع در مدل پیشنهادی هم جستجوی سراسری و هم جستجوی محلی به خوبی انجام می‌شوند. در AFSA، در هر تکرار برای تک‌تک ماهی‌های مصنوعی دو رفتار حرکت جمعی و دنباله‌روی اجرا می‌شوند که هر کدام می‌توانند به طور جداگانه موجب بهبود وضعیت ماهی‌های مصنوعی شوند. در پارهای اوقات و در برخی شرایط یکی از این دو رفتار مؤثرتر از دیگری عمل می‌کند. در واقع در بعضی تکرارها اجرای مجدد یکی از این دو رفتار به جای اجرای دیگری نتایج بهتری را برای ماهی مصنوعی اجرا کنده آن به دنبال خواهد داشت. در مدل پیشنهادی هر یک از ماهی‌های مصنوعی دارای یک اتوماتای یادگیر مربوط به خود هستند که در هر تکرار رفتارهای اجرایی را برای ماهی مصنوعی تعیین می‌کند. ادامه این مقاله به صورت زیر سازماندهی شده است: بخش دوم به معرفی الگوریتم AFSA استاندارد می‌پردازد، در بخش سوم اتوماتای یادگیر به اختصار معرفی شده است، در بخش چهارم مدل پیشنهادی شرح داده می‌شود، بخش پنجم نتایج آزمایشات را ارائه می‌کند و بخش پایانی به نتیجه‌گیری می‌پردازد.

۲- الگوریتم دسته ماهی‌های مصنوعی

در دنیای زیر آب، ماهی‌ها می‌توانند مناطقی را پیدا کنند که دارای غذای بیشتری است، که این امر با جستجوی فردی یا گروهی ماهی‌ها محقق می‌شود. بنابراین در دنیای زیر آب، معمولاً در جایی تعداد ماهی بیشتری وجود دارد که دارای غذای بیشتری است. مطابق با این ویژگی، مدل ماهی مصنوعی با رفتارهای جستجوی غذا، حرکت گروهی^۳ و دنباله‌روی^۴ ارائه شده است که به وسیله آنها فضای مسئله جستجو می‌شود [۷]. ماهی مصنوعی یک موجودیت ساختگی از ماهی واقعی می‌باشد که در تحلیل و تفسیر مسائل استفاده شده است.

می‌روند و به سرعت به غذا می‌رسند. اگر X_i وضعیت فعلی ماهی مصنوعی باشد، همسایه X_j را کاوش می‌کند ($d_{ij} \leq Visual$) و Y_j را می‌بیند، اگر $Z_j > Y_j$ و $n_j/n > \delta$ باشد یعنی وضعیت Z_j دارای تراکم غذایی بیشتری نسبت به وضعیت فعلی است و تراکم جمعیتی در نزدیکی آن زیاد نمی‌باشد، پس یک گام به سمت Z_j می‌رویم که با استفاده از فرمول ۵ محاسبه می‌گردد [۲]:

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{\|X_j - X_i^{(t)}\|} \cdot Step.Rand() \quad (5)$$

در غیر این صورت رفتار جستجوی غذا را اجرا می‌کند.

شبه کد AFSA در شکل ۲ مشاهده می‌شود. همانطور که در شبه کد شکل ۲ مشاهده می‌شود، نحوه اجرای رفتارهای AFSA برای تمام ماهی‌های مصنوعی بدین صورت است که در هر تکرار هردوی رفتارهای دنباله روی و حرکت جمعی برای هر ماهی مصنوعی اجرا می‌شود و در صورتی که اجرای هر کدام از این دو رفتار موفقیت آمیز نبود، رفتار جستجوی غذا برای ماهی مصنوعی اجرا می‌گردد.

```

For each Artificial Fish
    Initialize Artificial Fish Randomly
End For
Calculate Fitness Value of All Artificial Fishes
Do
    For each Artificial Fish
        Execute Swarm Behavior /*If not success Execute Prey
        Behavior*/
        Execute Follow Behavior /*If not success Execute Prey
        Behavior*/
    End For
While Max iteration

```

شکل ۲: شبکه کد الگوریتم AFSA

۳- اتوماتاهای یادگیر

اتوماتای یادگیر [۱۰]. مانشینی است که می‌تواند تعداد متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف اتومات را به حداقل برساند. اتوماتای یادگیر در تعامل با محیط، در انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداقل برساند. اتوماتای یادگیر در فرمول ۳ نشان داده شده است.



شکل ۳: ارتباط بین اتوماتای یادگیر و محیط.

گرایش به غذا یک رفتار اساسی زیستی است. در مجموع ماهی‌ها بوسیله بینایی یا حواسشان تراکم و غلظت غذا را در آب می‌فهمند سپس تصمیم می‌گیرند که به سمت آن حرکت کنند یا خیر. اگر X_i وضعیت فعلی ماهی مصنوعی باشد، یک وضعیت Z_j را به طور تصادفی در میدان دید ماهی مصنوعی انتخاب کنیم، $Y_j = f(X)$ برابر تراکم و غلظت غذا (مقدار تابع هدف) می‌باشد که با FC^5 نیز نشان داده می‌شود. وضعیت X_j با استفاده از فرمول ۱ بدست می‌آید [۲]:

$$X_j = X_i + Visual.Rand() \quad (1)$$

حال تراکم غذایی در Z_j را با تراکم غذایی در وضعیت فعلی مقایسه می‌کنیم، اگر $Z_j > Y_i$ باشد (در مسائلی بیشینه‌سازی)، ماهی مصنوعی مورد نظر از وضعیت فعلی اش یک گام به جلو در جهت Z_j حرکت می‌کند که با استفاده از فرمول ۲ محاسبه می‌شود:

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{\|X_j - X_i^{(t)}\|} \cdot Step.Rand() \quad (2)$$

اما در صورتی که $Z_j < Y_i$ نباشد، یک وضعیت Z_j دیگر را به با استفاده از فرمول ۱ انتخاب می‌کنیم و دوباره تراکم غذایی آن را می‌سنجم تا بهفهمیم که آیا شرط جلو رفتن را ارضاء می‌کند یا نه، اگر پس از try_number بار موفق به ارضاء کردن شرط جلو رفتن ($Z_j > Y_i$) نشدمیم، ماهی مصنوعی مورد نظر، یک گام به صورت آزادانه با استفاده از فرمول ۳ حرکت می‌کند:

$$X_i^{(t+1)} = X_i^{(t)} + Step.Rand() \quad (3)$$

۲-۲- رفتار حرکت گروهی ماهی‌ها

ماهی‌ها به طور طبیعی در گروه‌هایی جمع می‌شوند که موجب افزایش امنیت و بقای آنها می‌شود. X_c موقعیت مرکزی (X_{Center}) را نشان می‌دهد و n_f برابر تعداد همسایگان موجود در میدان دید X است ($d_{cx} \leq Visual$) و n برابر مجموع تعداد ماهی‌های مصنوعی می‌باشد. حال اگر Z_j تراکم غذا در موقعیت مرکزی باشد و $Z_j > Y_i$ باشد یعنی موقعیت وضعیت فعلی باشد، اگر $Z_j > Y_i$ باشد و $n_j/n > \delta$ باشد یعنی موقعیت مرکزی دارای تراکم غذایی بالاتری نسبت به وضعیت کنونی است و تراکم جمعیت در آن بالا نیست، پس یک گام به سمت میدان مرکزی پیش می‌رویم که با استفاده از فرمول ۴ محاسبه می‌گردد [۹ و ۲]:

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_c - X_i^{(t)}}{\|X_c - X_i^{(t)}\|} \cdot Step.Rand() \quad (4)$$

در غیر این صورت، رفتار جستجوی غذا اجرا می‌گردد.

۳-۲- رفتار دنباله‌روی

در فرآیند حرکت دسته‌ای ماهی‌ها، هنگامی که یک ماهی یا تعدادی از آنها غذا پیدا کنند، همسایگان و ماهی‌های نزدیک به آنها بدنیال آنها

وجود دارد که هر کدام از ماهی‌های مصنوعی دارای یک موقعیت در فضای مسئله می‌باشند.

در AFSA ماهی‌های مصنوعی در فضای میدان دید خود با اجرای رفتارهای AFSA به جستجوی فضای مسئله می‌پردازند و به اندازه درصدی تصادفی از طول گام به سمت هدف خود در هر تکرار حرکت می‌کنند. در AFSA استاندارد، تعیین مقدار اولیه این دو پارامتر تأثیر بسزایی در کیفیت نتیجه نهایی دارد. مقدار این دو پارامتر تا انتهای اجرای الگوریتم ثابت و برابر با همان مقدار اولیه آنها باقی می‌ماند. در صورتی که مقدار اولیه این دو پارامتر را بزرگ در نظر بگیریم دسته ماهی‌های مصنوعی می‌توانند با سرعت بیشتری به سمت بهینه سراسری حرکت کنند زیرا ماهی‌های مصنوعی می‌توانند فضای بزرگتر را در اطراف خود کاوش کنند و با طول گام بزرگتری در هر تکرار حرکت کنند. در این شرایط دسته ماهی‌های مصنوعی توانایی بیشتری برای عبور از بهینه‌های محلی خواهد داشت زیرا بزرگ بودن این دو پارامتر باعث گستردگی شدن دسته می‌شود و فاصله ماهی‌های مصنوعی از یکدیگر بیشتر خواهد بود. بدین ترتیب ماهی‌هایی که در بهینه محلی گرفتار شده‌اند، با رفتن به دنبال دیگر اعضای دسته که خارج از آن بهینه محلی هستند از آن خارج می‌شوند. همچنین بزرگ بودن پارامتر میدان دید باعث بزرگ شدن فضای جستجوی ماهی‌های مصنوعی می‌شود و آنها می‌توانند فاصله‌های دورتری را کاوش کنند که این امر سبب می‌شود بتوانند فضاهای خارج از بهینه‌های محلی که در آن گرفتار شده‌اند را جستجو و در نتیجه از آن خارج شوند. با این حال بزرگ بودن مقدار پارامترهای میدان دید و طول گام دارای معایبی می‌باشد، در این حالت دقت و پایداری الگوریتم در جستجوی محلی کاهش می‌یابد.

در واقع در این حالت الگوریتم جستجوی سراسری را به خوبی انجام می‌دهد ولی پس از اینکه دسته ماهی‌های مصنوعی به نزدیکی بهینه سراسری رسید، الگوریتم از توانایی انجام یک جستجوی محلی قابل قبول عاجز است زیرا در این حالت مقدار پارامتر میدان دید بزرگتر از فضایی است که لازم است ماهی مصنوعی آن را جستجو کند بنابراین احتمال یافتن مکان‌های با مقدار شایستگی بهتر کاهش می‌یابد همچنین به دلیل بزرگ بودن طول گام، ماهی‌های مصنوعی از بهینه سراسری عبور می‌کنند و حتی از آن دور می‌شوند که این امر باعث کاهش دقت و پایداری الگوریتم در نزدیکی بهینه سراسری می‌شود.

در صورتی که مقدار این دو پارامتر را کوچک در نظر بگیریم الگوریتم می‌تواند با پایداری و دقت بیشتری جستجوی محلی را انجام دهد ولی در این حالت ماهی‌ها با سرعت کمتری به سمت هدف حرکت می‌کنند و دسته آنها جمع‌تر و فشرده‌تر می‌شود که در نتیجه توانایی گذشتن از بهینه‌های محلی آنها کاهش می‌یابد. بنابراین می‌توان به این نتیجه رسید که برای به دست آوردن نتایج بهتر باید مقدار اولیه پارامترهای میدان دید و طول گام را بزرگ در نظر گرفت تا دسته ماهی‌های مصنوعی با سرعت بیشتری به سمت بهینه سراسری حرکت کنند و در مسیر رسیدن به آن با توانایی بالاتری از بهینه‌های محلی بگذرند. ولی

محیط را می‌توان توسط سه تابی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. هرگاه β مجموعه‌ای دوعضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q، (n) می‌تواند به طور گسسته یک مقدار از مقادیر موجود در بازه $[0, 1]$ باشد و در محیط از نوع S، $\beta(n)$ متغیری تصادفی در بازه $[0, 1]$ است. c احتمال اینکه عمل a_n نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند.

```
Initialize p to [1/s,1/s,...,1/s] /*s is the number of actions*/
While not done
    Select an action i based on the probability vector p
    Evaluate action and return a reinforcement signal β
    Update probability vector using learning rule
End While
```

شکل ۴: اتوماتای یادگیر با ساختار متغیر.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $LA \equiv \{\alpha, \beta, p, T\}$ نشان داد که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتومات، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتومات، $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از اعمال اتوماتای یادگیر و $T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل a_n در مرحله n ام انتخاب شود:

-پاسخ مطلوب

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n), \forall j \neq i \end{aligned} \quad (6)$$

-پاسخ نامطلوب

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= \left(\frac{b}{r-1}\right) + (1-b)p_j(n), \forall j \neq i \end{aligned} \quad (7)$$

در روابط ۶ و ۷، a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت مختلف را می‌توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} می‌نامیم، زمانیکه b از a خیلی کوچکتر باشد الگوریتم را $L_{R&P}$ می‌نامیم و زمانیکه b مساوی صفر باشد الگوریتم L_{RI} است [۱۴]. شکل ۴ شهود اتوماتای یادگیر را با ساختار متغیر نشان می‌دهد.

۴- مدل پیشنهادی

در این بخش یک الگوریتم دسته ماهی‌های مصنوعی بر پایه اتوماتای یادگیر که آنرا AFSA-LA می‌نامیم پیشنهاد می‌گردد. در مدل پیشنهادی همانند AFSA یک جمعیت از ماهی‌های مصنوعی

جمعی را برای ماهی مصنوعی مورد نظر در یک تکرار انجام داد. از سوی دیگر اگر یک ماهی مصنوعی در گروه دارای موقعیت و شایستگی مطلوبی باشد، اجرای رفتار حرکت جمعی نمی‌تواند برای آن مفید باشد، بنابراین برای آن، دو بار رفتار دنباله‌روی را به جای اجرای یکبار رفتار دنباله‌روی و حرکت جمعی اجرا می‌کنیم. برای این منظور به هر یک از ماهی‌های مصنوعی یک اتوماتای یادگیر اختصاص یافته که دارای دو عمل حرکت جمعی و دنباله‌روی است. این اتوماتای یادگیر در هر تکرار از اجرای الگوریتم تعیین می‌کند که ماهی مصنوعی مربوطه‌اش چه رفتاری را اجرا کند. در مدل پیشنهادی AFSA-LA، در هر تکرار، هر ماهی مصنوعی دو عمل از عمل‌های اتوماتای یادگیر مربوط به تنظیم رفتار را انجام می‌دهد. بدین ترتیب ماهی‌های مصنوعی هوشمند شده و در هر تکرار، بر اساس موقعیتی که در فضای مسئله و نسبت به دیگر اعضای گروه دارند، تصمیم می‌گیرند که چه رفتاری را انجام دهند.

```

For each Artificial Fish
    Initialize Artificial Fish Randomly
End For
Calculate Fitness Value of All Artificial Fishes
Do
    LA-P select an action
    Update Visual and Step according selected LA-P action
    For each Artificial Fish
        LA-B select an action ac
        Update Artificial Fish Position according ac
        /*if ac is Swarm Behavior, Execute Swarm Behavior else if
         ac is Follow Behavior, Execute Follow Behavior*/
        LA-B select an action ac again
        Update Artificial Fish Position according ac
        /*if ac is Swarm Behavior, Execute Swarm Behavior else if
         ac is Follow Behavior, Execute Follow Behavior*/
    End For
While Max iteration

```

شکل ۵: شبه کد الگوریتم AFSA

در مدل AFSA-LA، از یک اتوماتای یادگیر برای تنظیم پارامترهای ماهی‌های مصنوعی و از یک اتوماتای یادگیر دیگر برای تنظیم رفتارهای هر یک از ماهی‌های مصنوعی استفاده شده است. ساختار هر دو اتوماتای یادگیر به صورت متغیر است، محیط از نوع P در نظر گرفته شده است و از الگوریتم یادگیری شکل ۵ استفاده شده است. مقدار a و b با هم برابرند و الگوریتم از نوع LA_{RP} می‌باشد. شبه کد مدل AFSA-LA در شکل ۵ نشان داده شده است. در شبه کد شکل ۵، اتوماتای یادگیری که برای کنترل پارامترهای ماهی‌های مصنوعی به کار رفته با LA-P و اتوماتای یادگیری که برای تنظیم رفتارهای ماهی‌های مصنوعی به کار رفته را LA-B نامگذاری شده است. نتایج آزمایشات در بخش بعد ارائه شده است.

۵- نتایج آزمایشات

آزمایشات بر روی پنجتابع استاندارد صورت گرفته است که معمولاً به عنوان معیار سنجش الگوریتم‌های بهینه‌سازی در فضاهایی پیوسته و

مقدار این دو پارامتر باید به طور تطبیقی در طول اجرای الگوریتم کاهش یابد تا هنگامی که دسته ماهی‌های مصنوعی به محدوده بهینه سراسری رسید، بتواند با دقت بیشتری محدوده بهینه سراسری را برای این دو مقادیر بهتر کاوش کند.

برای کنترل مقدار پارامترهای میدان دید و طول گام و برقراری موازنۀ میان جستجوی سراسری و جستجوی محلی، یک اتوماتای یادگیر به اضافه می‌گردد. در این مدل، اتوماتای یادگیر اندازه پارامترهای میدان دید و طول گام را در طول اجرا تنظیم می‌کند. در واقع اتوماتای یادگیر در هر تکرار تعیین می‌کند که اندازه این دو پارامتر: برابر با مقدار آنها در تکرار قبل باشد. کوچکتر از مقدار آنها در تکرار قبل باشد. بزرگتر از مقدار آنها در تکرار قبل باشد.

بنابراین این اتوماتای یادگیر سه عمل مختلف کاهش، افزایش و بدون تغییر گذاشت پارامترهای میدان دید و طول گام را در هر تکرار از اجرای الگوریتم انجام می‌دهد. بنابراین هر یک از سه عمل اتوماتای یادگیر اندازه این دو پارامتر را نسبت به مقدارشان در تکرار قبل تعیین می‌کنند. در عمل اول، مقدار این دو پارامتر در تکرار جاری برابر مقدار آنها در تکرار قبل در نظر گرفته می‌شود. در عمل دوم مقدار این دو پارامتر در تکرار جاری از ضرب کردن مقدار پیشین آنها در بک عدد بزرگتر از یک بدست می‌آید، بدین ترتیب با اجرای عمل دوم اتوماتای یادگیر مقدار این دو پارامتر افزایش می‌یابد. در عمل سوم، برای کاهش مقدار پارامترهای میدان دید و طول گام، مقدار آنها در تکرار پیشین را یک عدد کوچکتر از یک ضرب می‌کند که این عدد به صورت تطبیقی و بر اساس شماره تکرار جاری توسط فرمول ۸ تولید می‌شود.

$$W_{itr} = X + \frac{(itr_{max} - itr)^n}{(itr_{max})^n} \times (Y - X) \quad (8)$$

در فرمول ۸، n شاخص غیرخطی کننده است، در این فرمول، وزن مربوط به تکرار itr ام، بر اساس شماره تکرار جاری و تکرار نهایی در بازه [X, Y] تعیین می‌شود. فرمول ۸ به گونه‌ای طراحی شده است که خروجی آن در طول تکرارهای اجرای الگوریتم از Y تا X به صورت غیر خطی کاهش می‌یابد.

در هر تکرار از اجرای الگوریتم، دو رفتار حرکت جمعی و دنباله‌روی برای تک‌تک ماهی‌های مصنوعی اجرا می‌شود. با اجرای هر یک از این دو رفتار، هر کدام از ماهی‌های مصنوعی بر اساس موقعیتی که در فضای مسئله و دیگر اعضای گروه دارند می‌توانند موقعیت خود را بهبود بخشدند. بدیهی است که در پاره‌ای اوقات اجرای دو مرتبه یکی از این دو رفتار می‌تواند مفیدتر از اجرای یک بار هر یک از این دو رفتار با هم باشد. برای مثال برای یک ماهی مصنوعی که از گروه دور افتاده، اجرای رفتار دنباله‌روی نمی‌تواند مفید باشد زیرا این ماهی مصنوعی دارای همسایه‌ای در میدان دید خود نیست، بنابراین می‌توان به جای یکبار اجرای رفتارهای دنباله‌روی و حرکت جمعی، دوبار رفتار حرکت

ماهی‌های مصنوعی توسط اتماتای یادگیر، سرعت و دقت ماهی‌های مصنوعی در حرکت به سمت بهینه سراسری افزایش یافته است. نتایج شبیه‌سازی‌ها نشان می‌دهد که الگوریتم پیشنهاد شده AFSA-LA در مجموع دارای کارایی بهتری نسبت به AFSA استاندارد و IAFSA می‌باشد.

جدول ۲: نتایج مقایسه AFSA استاندارد، IAFSA و AFSA-LA بر روی تابع روزنبراک.

بدترین	متوسط	بهترین	الگوریتم	ابعاد
۲۰۶,۲۲۵۵	۳۳,۶۹۰۶	۷,۵۰۴۱	StdAFSA	۱۰
۸,۷۳۷۰	۶,۹۵۲۲	۴,۳۷۰۴	IAFSA	
۶,۵۵۱۸	۱,۱۳۴۳	۰,۱۸۶۶	AFSA-LA	
۵۵۶,۴۶۹۴	۷۲,۸۹۱۱	۱۸,۷۱۴۲	StdAFSA	۲۰
۲۰,۰۱۴۴	۱۸,۲۴۳۶	۱۵,۳۸۵۸	IAFSA	
۱۹,۴۹۱۶	۱۷,۹۸۲۴	۱۴,۰۹۱۰	AFSA-LA	
۴۳۶,۱۴۶۶	۱۰,۸,۲۹۸۱	۴۵,۸۱۴۲	StdAFSA	۳۰
۱۰۰,۴۸۱۸	۴۷,۸۲۹۱	۲۶,۱۱۹۴	IAFSA	
۲۸,۶۹۸۴	۲۷,۲۳۷۲	۲۴,۸۲۸۵	AFSA-LA	

جدول ۳: نتایج مقایسه AFSA استاندارد، IAFSA و AFSA-LA بر روی تابع آکلی.

بدترین	متوسط	بهترین	الگوریتم	ابعاد
۲,۶۳۹۵	۱,۱۶۵۹	۰,۱۶۹۸	StdAFSA	۱۰
۰,۲۷e-۱۴	۰,۲۶e-۱۴	۰,۲۶e-۱۴	IAFSA	
۲,۶۵e-۱۵	۲,۶۵e-۱۵	۲,۶۴e-۱۵	AFSA-LA	
۳,۶۶۵۴	۲,۴۸۹۶	۰,۴۹۱۶	StdAFSA	۲۰
۰,۱۳e-۱۳	۰,۰۸e-۱۳	۰,۰۶e-۱۳	IAFSA	
۱,۳۳e-۱۴	۷,۱۰e-۱۵	۶,۲۱e-۱۵	AFSA-LA	
۶,۷۷۲۳	۴,۶۵۱۸	۲,۹۰۲۴	StdAFSA	۳۰
۱,۰۵۹۷	۰,۵۴۷۶	۰,۰۰۰۰۸	IAFSA	
۲,۷۵e-۱۴	۱,۸۴e-۱۴	۱,۳۳e-۱۴	AFSA-LA	

جدول ۴: نتایج مقایسه AFSA استاندارد، IAFSA و AFSA-LA بر روی تابع رستربیجن.

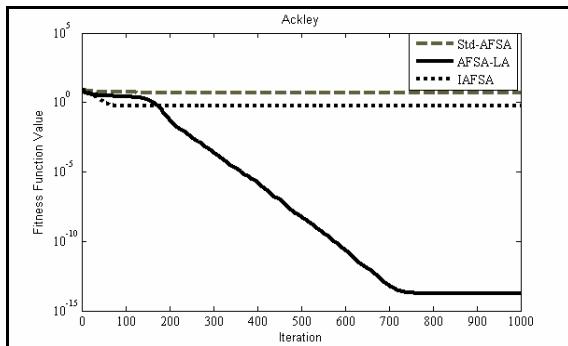
بدترین	متوسط	بهترین	الگوریتم	ابعاد
۱۶,۱۶۹۷	۸,۱۸۴۱	۱,۰۱۷۹	StdAFSA	۱۰
۲,۹۹۴۹	۰,۲۴۹۷	۰	IAFSA	
۱,۹۸۹۹	۰,۱۴۹۲	۰	AFSA-LA	
۴۸,۲۵۵۰	۱۱,۳۱۹۸	۲,۰۵۴۸	StdAFSA	۲۰
۲,۹۸۴۸	۱,۴۶۵۰	۰	IAFSA	
۲,۹۸۴۸	۰,۴۹۷۴	۰	AFSA-LA	
۴۴,۳۹۷۱	۱۸,۲۵۰۷	۵,۱۱۸۳	StdAFSA	۳۰
۶,۹۶۴۷	۲,۳۲۹۲	۰,۱۸e-۱۳	IAFSA	
۰,۹۹۴۹	۰,۰۹۹۴	۰	AFSA-LA	

ایستا مورد استفاده قرار می‌گیرند. توابع استفاده شده عبارتند از توابع روزنبراک، آکلی، رستربیجن، اسفیر و گریوانک که در جدول ۱ نشان داده شده‌اند. شایان ذکر است که مقدار بهینه تمام این توابع برابر با صفر است.

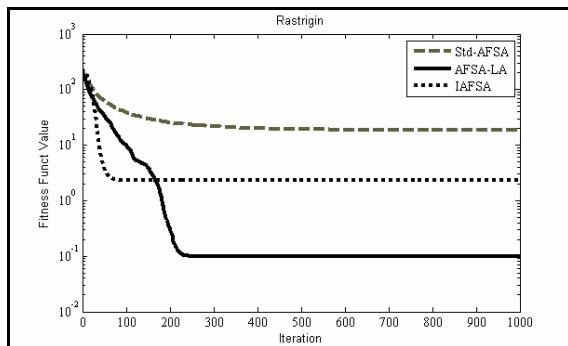
جدول ۱: توابع استاندارد روزنبراک، آکلی، رستربیجن، اسفیر و گریوانک.

نام تابع	متغیرها	محدوده	معادله تابع
روزنبراک		±۵۰	$\sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$
آکلی		±۳۲	$20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$
رستربیجن		±۵,۱۲	$\sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$
اسفیر		±۱۰۰	$\sum_{i=1}^n x_i^2$
گریوانک		±۶۰۰	$\sum_{i=1}^n \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

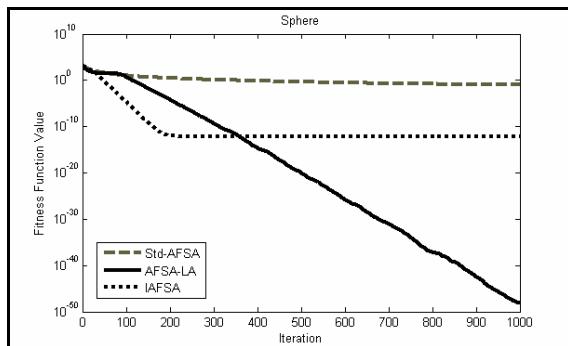
آزمایشات در فضاهای ۱۰، ۲۰ و ۳۰ بعدی انجام شده است. مقدار اولیه پارامتر میدان دید برابر با ۵۰ درصد و مقدار پارامتر طول گام برابر ۳۰ درصد از محدوده متغیرهای تابع شایستگی در نظر گرفته می‌شود. حداقل تعداد تکرار ۱۰۰۰ می‌باشد، فاکتور شلوغی برابر ۰/۵، حداقل تعداد تلاش‌ها برابر ۱۰ است و تعداد ماهی‌ها برابر ۳۰ می‌باشد. ضرایب پاداش و جریمه برای اتماتاهای یادگیر در AFSA-LA برابر ۰,۰۱ در نظر گرفته شده است. برای عمل دوم اتماتای یادگیر مربوط به کنترل پارامترهای میدان دید و طول گام مقداری عددی که در این پارامترها ضرب می‌شود ۱,۰۵ در نظر گرفته شده است. برای عمل سوم همان اتماتای یادگیر، مقدار X ، Y در رابطه ۸ به ترتیب برابر ۰,۹، ۰,۵ و ۱ در نظر گرفته شده است. در آزمایشات از AFSA همراه با پارامترهای تطبیقی (IAFSA) نیز استفاده شده است. در این AFSA در هر تکرار مقدار پارامترهای میدان دید و طول گام در یک ثابت مشتب کوچکتر از یک ضرب می‌شود که باعث می‌شود مقدار این دو پارامتر در هر تکرار نسبت به مقدار آنها در تکرار قبل کوچکتر شوند. که در نهایت با بزرگ در نظر گرفتن مقدار اولیه این دو پارامتر، هم توانایی جستجوی محلی و هم توانایی جستجوی سراسری این الگوریتم بالا می‌رود. در آزمایشات مقداری که در هر تکرار در IAFSA در مقدار پارامترهای طول گام و میدان دید ضرب می‌شود را برابر ۰/۹ در نظر می‌گیریم. آزمایشات ۳۰ بار تکرار شده‌اند و متوسط نتایج، بهترین و بدترین آنها در جداول ۲ تا ۵ در ابعاد ۱۰، ۲۰ و ۳۰ آورده شده است. همانطور که در جداول ۲ تا ۶ مشاهده می‌شود، نتایج بدست آمده از مدل پیشنهادی AFSA بهتر از AFSA-LA استاندارد است، در واقع در الگوریتم پیشنهاد شده، با کنترل پارامترهای میدان دید و طول گام توسط اتماتای یادگیر، الگوریتم قادر است هم جستجوی سراسری و هم جستجوی محلی را به خوبی انجام دهد همچنین با تنظیم رفتارهای انجام شده بوسیله



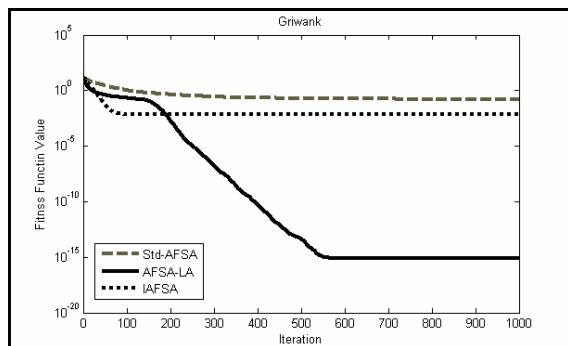
شکل ۷: مقایسه متوسط نتایج اجرای AFSA استاندارد، و IAFSA و AFSA-LA در ۳۰ بار اجرا بر روی تابع آکلی ۳۰ بعدی.



شکل ۸: مقایسه متوسط نتایج اجرای AFSA استاندارد، و IAFSA و AFSA-LA در ۳۰ بار اجرا بر روی تابع رستریجن ۳۰ بعدی.



شکل ۹: مقایسه متوسط نتایج اجرای AFSA استاندارد، و IAFSA و AFSA-LA در ۳۰ بار اجرا بر روی تابع اسپیر ۳۰ بعدی.



شکل ۱۰: مقایسه متوسط نتایج اجرای AFSA استاندارد، و IAFSA و AFSA-LA در ۳۰ بار اجرا بر روی تابع گریوانک ۳۰ بعدی.

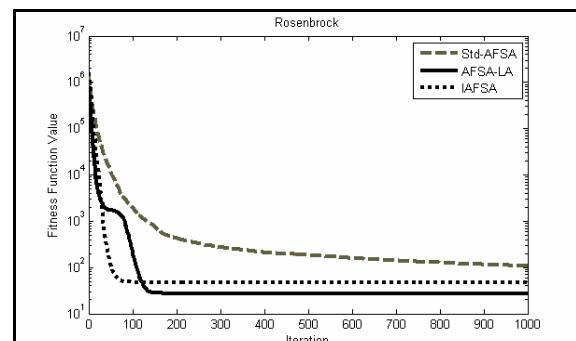
جدول ۵: نتایج مقایسه AFSA استاندارد، و IAFSA و AFSA-LA بر روی تابع اسپیر.

الگوریتم	بعده	بهترین	متوسط	بدترین
StdAFSA	۱۰	۰,۰۲۰۱	۰,۰۰۶۹	۰,۰۲۹۹
IAFSA		۳,۲۸e-۹۰	۱,۰۹e-۸۹	۲,۳۹e-۸۹
AFSA-LA		۴,۲۵e-۷۸	۴,۶۴e-۷۴	۴,۲۴e-۷۳
StdAFSA	۲۰	۰,۰۰۸۵	۰,۰۲۴۵	۰,۰۴۰۸
IAFSA		۱,۴۷e-۸۸	۲,۰۲e-۸۸	۲,۷۹e-۸۸
AFSA-LA		۶,۳۴e-۶۶	۸,۰۳e-۶۱	۱,۱۶e-۵۹
StdAFSA	۳۰	۰,۰۵۹۴	۰,۰۹۳۸	۰,۱۸۵۷
IAFSA		۱,۳۹e-۲۴	۰,۰۸e-۱۱	۰,۶۵e-۱۱
AFSA-LA		۵,۳۴e-۵۳	۷,۰۵e-۴۹	۶,۶۴e-۴۸

جدول ۶: نتایج مقایسه AFSA استاندارد، و IAFSA و AFSA-LA بر روی تابع گریوانک.

الگوریتم	بعده	بهترین	متوسط	بدترین
StdAFSA	۱۰	۰,۰۰۱۳	۰,۰۲۴۴	۰,۰۴۰۷
IAFSA		•	۷,۸۵e-۱۲	۱,۱۵e-۱۰
AFSA-LA		•	۳,۳۳e-۱۷	۲,۲۲e-۱۶
StdAFSA	۲۰	۰,۰۰۳۱	۰,۰۶۴۱	۰,۲۰۱۳
IAFSA		۵,۰۵e-۰۷	۶,۴۲e-۰۵	۳,۲۰e-۰۴
AFSA-LA		۲,۲۱e-۱۶	۲,۳۳e-۱۶	۴,۴۴e-۱۶
StdAFSA	۳۰	۰,۰۰۷۶	۰,۱۵۸۲	۰,۳۰۷۹
IAFSA		۰,۰۰۱۹	۰,۰۰۷۳	۰,۰۱۷۱
AFSA-LA		۲,۲۲e-۱۶	۸,۳۲e-۱۶	۶,۴۳e-۱۵

همانطور که در جداول ۲ تا ۶ مشاهده می‌شود، عملکرد AFSA-LA در تمام موارد به غیر از ابعاد پایین تابع اسپیر، عملکرد بهتری نسبت به AFSA استاندارد و IAFSA دارد. در AFSA-LA چون کاهش مقدار پارامترهای میدان دید و طول گام توسط LA انجام می‌شود و همچنین ماهیهای مصنوعی به کمک LA مربوط به خود به صورت هوشمندانه‌تری رفتارهای خود را انتخاب می‌کنند، این روش در مجموع دارای کارایی بهتری از دو روش دیگر است.



شکل ۱۱: مقایسه متوسط نتایج اجرای AFSA استاندارد، و IAFSA و AFSA-LA در ۳۰ بار اجرا بر روی تابع روزنبراک ۳۰ بعدی.

- [8] M. Jiang, Y. Wang, F. Rubio and D. Yuan, “*Spread Spectrum Code Estimation by Artificial Fish Swarm Algorithm*”, In Proc. IEEE International Symposium on Intelligent Signal Processing (WISP), 2007.
- [9] M. Y. Jiang and D. F. Yuan, “*Artificial Fish Swarm Algorithm and Its Applications*”, In Proc. of the International Conference on Sensing, Computing and Automation (ICSCA'2006), 2006.
- [10] K. S. Narendra and M. A. L. Thathachar, “*Learning Automata: An Overview*”, Prentice Hall, 1989.
- [11] M. R. Meybodi and H. Beigy, “*A Note on Learning Automata Based Schemes for Adaptation of BPPParameters*”, Journal of Neurocomputing, 2002.
- [12] M. N. Howell, T. J. Gordon, and F. V. Brandao, “*Genetic Learning Automata for Function Optimization*”, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 2002.
- [13] M. Sheybani and M. R. Meybodi, “*PSO-LA: A New Model for Optimization*”, In Proc. of CSICC06, 2006.
- [14] M. A. L. Thathachar and P. S. Sastry, “*Varieties of Learning Automata: An Overview*”, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, 2002.

¹ Artificial Fish Swarm Algorithm

² AF Prey

³ AF Swarm

⁴ AF Follow

⁵ Food Consistence

⁶ Learning Automata

در شکل‌های ۶ تا ۱۰، نمودار مقدار تابع شایستگی را در طول ۱۰۰۰ تکرار از اجرای AFSA استاندارد و مدل پیشنهادی AFSA-LA برای توابع استاندارد رستربیجن، آکلی، روزنبراک، اسفیر و گریوانک نشان داده شده‌اند.

۶- نتیجه‌گیری

در این مقاله یک مدل جدید برای بهینه‌سازی بر پایه دو مدل الگوریتم دسته‌مهی‌های مصنوعی و اتوماتاهای یادگیر پیشنهاد گردید. در مدل پیشنهادی جمعیت به یک اتوماتای یادگیر برای کنترل مقدار پارامترهای میدان دید و طول گام مجهز می‌باشد. به وسیله این اتوماتای یادگیر، مدل پیشنهادی قادر است که مقدار این دو پارامتر را در طول تکرارهای اجرای الگوریتم با موقعیت گروه در فضای مسئله تطبیق دهد. این اتوماتای یادگیر دارای سه عمل کاهش، افزایش و بدون تغییر گذاشت پارامترهای میدان دید و طول گام نسبت به مقدار آنها در تکرار پیشین است که بوسیله این اعمال مقدار این دو پارامتر را در طول اجرا کنترل می‌کند. بدین ترتیب در مدل پیشنهادی بین جستجوی سراسری و جستجوی محلی موازن برقرار شده است و هر دوی آنها به خوبی انجام می‌شوند. در مدل پیشنهادی، هر یک از ماهی‌های مصنوعی نیز به یک اتوماتای یادگیر مجهز شده‌اند که به مثابه مغز متغیر و کنترل‌کننده حرکات و رفتارهای آنها عمل می‌کند. این اتوماتای یادگیر دارای دو عمل حرکت جمعی و دنباله‌روی است که نحوه حرکت ماهی مصنوعی را تعیین می‌کند. نتایج شبیه‌سازی‌ها نشان داد که الگوریتم پیشنهاد شده دارای کارایی بهتری نسبت به AFSA استاندارد است.

مراجع

- [1] Y. Liu and K. M. Passino, “*Swarm Intelligence: A Survey*”, In Proc. of 4th International Conference of Swarm Intelligence , 2005.
- [2] 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, 2002.
- [3] M. Jiang, Y. Wang, S. Pfletschinger, M. A. Lagunas and D. Yuan , “*Optimal Multiuser Detection with Artificial Fish Swarm Algorithm*”, In Proc. of International Conference on Intelligent Computing (ICIC), 2007.
- [4] G. Cui, X. Cao, J. Zhou and Y. Wang, “*The Optimization of DNA Encoding Sequences Based on Improved AFS Algorithms*”, In Proc. of IEEE International Conference on Automation and Logistics, Jinan, 2007.
- [5] H. Chen, S. Wang, J. Li and Y. Li, “*A Hybrid of Artificial Fish Swarm Algorithm and Particle Swarm Optimization for Feed-forward Neural Network Training*”, In Proc. of New Weaponry Technology & Application, 2006.
- [6] 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 4th International Conference on Machine Learning and Cybernetics, Guangzhou, 2005.
- [7] J. M. Xiao, X. M. Zheng and X. H. Wang, “*A Modified Artificial Fish Swarm Algorithm*”, In Proc. of the IEEE 6th World Congress on Intelligent Control and Automation, Dalian, 2006.